

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Галузь знань 12 – Інформаційні технології

Спеціальність 126 – Інформаційні системи та технології

на тему: «Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів»

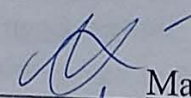
КвРІСТ.180238.22.01.07 ПЗ

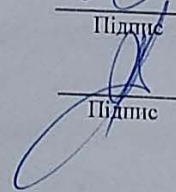
Виконав: студент 2 курсу, група ІСТм-22-1

Керівник: доктор техн. наук, професор
Науковий ступінь, вчене звання

До захисту допускаю:
Зав. кафедри КІС, д.т.н., проф.

Т.О. Говорущенко
18 12 2023 р.


Підпис Мандрик А.І.
Ініціали, прізвище


Підпис Лисенко С.М.
Ініціали, прізвище

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

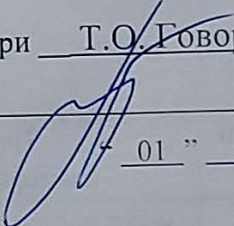
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорушенко



01 " 04 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Мандрику Андрію Ігоровичу

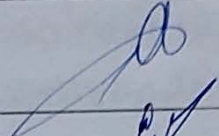
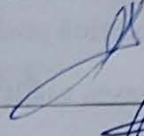
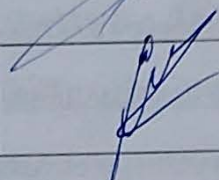
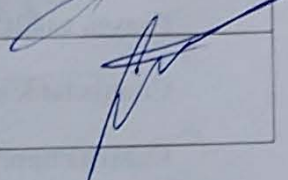
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проектів
2. Керівник проекту (роботи) Лисенко С.М., д.т.н., професор
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.08.2023 р. №30

2. Строк подання студентом проекту (роботи) на кафедру 10.12.2023 р.
3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Дослідження методів та засобів інформаційної технології інтелектуалізованого планувальника роботи команди розробників проектів
Обґрунтування і вибір теоретичних та експериментальних методів дослідження
Розроблення удосконаленого методу інтелектуалізованого планувальника роботи команди розробників проектів
Технологія інтелектуалізованого планувальника роботи команди розробників проектів
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

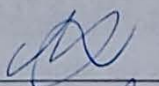
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІС		

7. Дата видачі завдання « 03 » 04 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проєкту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	03.04.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	03.05.2023	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	03.06.2023	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	03.07.2023	виконано
5	Робота над науковою статтею	01.10.2023	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	01.10.2023	виконано
7	Робота над розділом 4 – проєктування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.11.2023	виконано
8	Оформлення пояснювальної записки згідно вимог	15.11.2023	виконано
9	Попередній захист ДРМ	16.11.2023	виконано
10	Захист ДРМ на засіданні ЕК	До 20.12.2023	

Студент

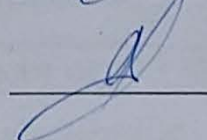


Підпис

А.І. Мандрик

Ініціали, прізвище

Керівник роботи



Підпис

С.М. Лисенко

Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.

Автор роботи: Мандрик Андрій Ігорович

Керівник роботи: доктор техн. наук, професор Лисенко С.М.

Пояснювальна записка: 82 с., 22 рис., 10 табл., 2 дод., 82 джерела.

Перелік ключових слів: Планування проєктів, командоутворення, генетичний алгоритм, оптимізація, інструменти візуалізації.

Об'єктом дослідження є процес оптимізації планувальна роботи команд розробників проєктів.

Предметом дослідження є метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.

Метою кваліфікаційної роботи є підвищення ефективності роботи планувальника роботи команд розробників ПЗ із застосуванням апарат еволюційних алгоритмів.

Для розв'язання поставлених задач використовувалися методи: аналітичні та математичні методи дослідження в галузі оптимізації; теорія генетичних алгоритмів та їх удосконалення; статистичні методи аналізу ефективності алгоритмів; сучасні програмні засоби для моделювання генетичних алгоритмів; комп'ютерні технології для проведення експериментів.

Наукова новизна роботи:

– набув подальшого розвитку метод побудови інтелектуалізованого планувальника роботи команди розробників проєктів, який на відміну від відомих базується на оптимізаційних процесах, які ґрунтуються на застосуванні еволюційних алгоритмів, і уможливорює підвищити ефективність роботи планувальника роботи команд розробників ПЗ.

– набули подальшого розвитку засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.

Практичне значення роботи полягає в тому, що запропонована інформаційна технологія інтелектуалізованого планувальника роботи команди розробників проєктів може бути використаний як складова частина проєктів де необхідний розподіл робочих процесів між працівниками.

ЗМІСТ

ВСТУП	6
1 ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ РОЗРОБНИКІВ ПРОЄКТІВ	9
1.1 Проблема оптимізації та планування.....	9
1.2 Відомі методи та засоби для оптимізації планування.....	10
1.3 Відомі засоби для оптимізації процесу планування проєктів	13
1.3.1 GanttProject.....	17
1.3.2 ProjectLibre.....	18
1.3.3 OpenProject.....	21
1.4 Постановка задачі дослідження.....	23
2 ОБҐРУНТУВАННЯ І ВИБІР ТЕОРЕТИЧНИХ ТА ЕКСПЕРИМЕНТАЛЬНИХ МЕТОДІВ ДОСЛІДЖЕННЯ	25
2.1 Теоретичні основи генетичних алгоритмів у плануванні проєктів та формуванні команд	25
2.2 Підготовка даних та обробка генетичним алгоритмом для оптимального планування проєкту	29
2.3 Структура інформаційної системи	33
2.4 Висновки	41
3 РОЗРОБЛЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ РОЗРОБНИКІВ ПРОЄКТІВ	42
3.1 Основи удосконаленого методу інтелектуалізованого планувальника роботи команди розробників проєктів.....	42
3.2 Етап збору та попередньої обробки вхідних даних.....	44

3.3	Етап застосування генетичного для завдання оптимізації	48	
3.4	Етап відображення та інтерпретації даних отриманих від генетичного алгоритму	58	
3.5	Експериментальні дослідження методу.....	62	
3.6	Висновки	65	
4 ЗАСОБИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ			
ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ			
РОЗРОБНИКІВ ПРОЄКТІВ.....			67
4.1	Архітектура інтелектуалізованого планувальника роботи команди розробників проєктів	67	
4.3	Вимоги до інтелектуалізованого планувальника роботи команди розробників проєктів	77	
4.4	Огляд розробленого інтерфейсу додатку для інтелектуалізованого планувальника роботи команди розробників проєктів	79	
4.5	Висновки.....	84	
ВИСНОВКИ.....			85
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ			87
ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....			95
ДОДАТОК Б КОПІЯ ОПУБЛІКОВАНОЇ СТАТТІ			98
ДОДАТОК В КОПІЯ ПРЕЗЕНТАЦІЇ.....			104

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – прикладний програмний інтерфейс

UI – візуальний інтерфейс

UX – користувацький досвід

DEAP – Розподілений еволюційний алгоритм на Python

PM – Управління проектами

AI – Штучний інтелект

ML – Машинне навчання

SDLC – Цикл розробки програмного забезпечення

ВСТУП

У сучасному світі управління проектами та командами стає все більш важливим для ефективності та успішності реалізації проектів. Ключовим завданням у цій сфері є розробка планувальників робіт, які оптимізують процеси, ефективно розподілять ресурси та підвищують продуктивність команди. Інтелектуальний планувальник робіт - це інструмент, який використовує методи та засоби інформаційних технологій для автоматизації планування, моніторингу та управління командами розробників. Ефективність таких систем безпосередньо впливає на швидкість реалізації проекту, якість продукту та здатність команди адаптуватися до змін у проекті.

Тому розробка та вдосконалення інформаційно-технологічних методів та інструментів для інтелектуальних планувальників робіт є актуальною та важливою задачею. Актуальність роботи полягає у дослідженні сучасних підходів до інтелектуалізації процесів планування, а також у розробці ефективних інструментів управління роботою команд розробників.

Об'єктом дослідження є процес оптимізації планувальна роботи команд розробників ПЗ.

Предметом дослідження є метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проектів.

Метою кваліфікаційної роботи є підвищення ефективності роботи планувальника роботи команд розробників проектів із застосуванням апарат еволюційних алгоритмів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити методи та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проектів
- дослідити проблему оптимізації планування;
- проаналізувати відомі засоби та методи оптимізації;
- обґрунтувати вибір теоретичних та експериментальних методів для дослідження;

- підготувати дані для обробки генетичним алгоритмом;
- створити структуру інформаційної технології;
- розробити удосконалений метод інтелектуалізованого планувальника роботи команди розробників проєктів;
- провести експериментальне дослідження методу;
- розробити архітектуру інтелектуалізованого планувальника роботи команди розробників проєктів;
- скласти вимоги до інтелектуалізованого планувальника роботи команди розробників проєктів;
- провести оцінку ефективності розробленого алгоритму.

Методи дослідження. У роботі застосовано наступні теорії та засоби:

- аналітичні та математичні методи дослідження в галузі оптимізації;
- теорія генетичних алгоритмів та їх удосконалення;
- статистичні методи аналізу ефективності алгоритмів;
- сучасні програмні засоби для моделювання генетичних алгоритмів;
- комп'ютерні технології для проведення експериментів.

Наукова новизна роботи:

– набув подальшого розвитку метод побудови інтелектуалізованого планувальника роботи команди розробників проєктів, який на відміну від відомих базується на оптимізаційних процесах, які ґрунтуються на застосуванні еволюційних алгоритмів, і уможливорює підвищити ефективність роботи планувальника роботи команд розробників ПЗ.

– набули подальшого розвитку засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.

Практичне значення роботи полягає в тому, що запропонована інформаційна технологія інтелектуалізованого планувальника роботи команди розробників проєктів може бути використаний як складова частина проєктів де необхідний розподіл робочих процесів між працівниками.

Рекомендації з використання результатів роботи. Отримані результати можуть бути використані у розробці програмного забезпечення, для оптимізації процесів в команді при реалізації проєктів різного призначення.

Важливість роботи і висновки. Робота має значення для розвитку методів оптимізації та їх застосування у різноманітних сферах.

Публікації. За темою кваліфікаційної роботи опубліковано одну статтю у фаховому виданні [1] та одні тези [2].

Структура та об'єм кваліфікаційної роботи. Кваліфікаційна робота складається з вступу, чотирьох розділів, висновку та додатків, її повний зміст 110 сторінок, основний зміст викладено на 86 сторінках, 2-х додатках, містить 22 рисунка, 10 таблиць, включає 82 найменування вітчизняної та зарубіжної літератури.

1 ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ РОЗРОБНИКІВ ПРОЄКТІВ

1.1 Проблема оптимізації та планування

Сфера програмної інженерії постійно стикається з роблемою невдач програмних проєктів. Як зазначено у звіті Standish Group CHAOS Report [3] за 2020 рік, прогноз для програмних проєктів не є обнадійливим. Більшість проєктів або перевищують свої бюджети, або не вкладаються у встановлені терміни, або не виконують необхідних функціональних можливостей. Шокуючим фактом є те, що майже чверть усіх проєктів припиняються на півдорозі або не мають жодної користі після завершення, що підкреслює різке зниження рівня успішності проєктів з роками [4].

Це спонукає до глибокого аналізу динаміки управління проєктами. Виникають ключові питання, що вказують на здібності команди і компетентність керівника проєкту в питаннях планування і розподілу ресурсів [5, 6]. У фокусі уваги опиняються два важливі аспекти: планування проєкту та укомплектування команди. Ці завдання, ключові для успіху програмного проєкту, тісно переплітаються з можливостями людських ресурсів [7].

В основі управління програмними проєктами лежить складне завдання визначення дорожньої карти проєкту. Це передбачає визначення результатів, оцінку зусиль, необхідних для кожного компонента, і побудову гнучкого, але надійного графіка проєкту [8-9]. Це не просто одноразове завдання; змінна природа програмних проєктів, схильна до мінливих вимог і непередбачуваних викликів, вимагає постійного перегляду розкладу, щоб забезпечити його відповідність поточному стану проєкту [10].

Традиційний підхід до формування команди, який часто базується на наявності ресурсів або ієрархічних міркуваннях, має тенденцію не враховувати цю складну динаміку. У багатьох випадках команди, сформовані без глибокого аналізу, можуть призвести до непорозумінь, невідповідності наборів навичок і

навіть міжособистісних конфліктів [11]. Такі сценарії не лише затримують проекти, але й можуть призвести до збільшення витрат, особливо коли ключові етапи доводиться переглядати через пропущені помилки або неправильно поставлені цілі.

1.2 Відомі методи та засоби для оптимізації планування

Було досліджено низку математичних методів для підвищення ефективності та результативності планування та розподілу ресурсів. Ці методи, кожен з яких має свої особливості застосування та переваги, стали предметом широких досліджень і практичного застосування в різних галузях.

Лінійне програмування, фундаментальний підхід в оптимізації, був використаний у дослідженні [12,13] щодо його застосування в плануванні робочої сили. Одне з досліджень включало розробку ЛП-моделі для оптимізації графіків змін у великій мережі роздрібної торгівлі. Модель мала на меті мінімізувати витрати на персонал, одночасно задовольняючи попит клієнтів та вподобання працівників.

Результати опублікованих досліджень, продемонстрували значну економію коштів і підвищення рівня задоволеності працівників, що підкреслило потенціал ЛП в управлінні трудовими ресурсами [14]. Оцінка сумісності між членами команди, C , може бути виражена наступним чином:

$$C(T) = \sum_{i=1}^n \sum_{j=i+1}^n (m_i, m_j), \quad (1.1)$$

де T - команда, що складається з членів m_i, m_j , а n - кількість членів команди. Цей алгоритм показав ефективність у формуванні збалансованих і згуртованих команд, але висвітлив проблему кількісної оцінки суб'єктивних факторів, таких як сумісність команди.

Системи вдосконаленого планування та складання графіків застосовують алгоритми оптимізації для подолання обмежень фіксованої логіки, сприяючи створенню більш реалістичного плану [15, 16], який збалансовує як ресурси, так і матеріали, таким чином покращуючи виробниче планування у виробничих умовах. Ці інтелектуальні підходи до оптимізації спрямовані на мінімізацію виробничого графіка за допомогою математичної моделі, що призводить до ефективного планування та оцінювання складних виробничих процесів [17].

Цілочисельне програмування ще один метод вирішення складних дискретних оптимізаційних задач. Ілюстративним прикладом є його застосування в авіаційній галузі для складання розкладу руху екіпажів. Дослідники розробили модель ЦП, яка ефективно призначала екіпажі на рейси, дотримуючись суворих нормативних і контрактних обмежень. Модель, яка була прийнята кількома авіакомпаніями, значно знизила операційні витрати і покращила використання екіпажів.

Програмування з обмеженнями широко застосовується в освітніх розкладах.

Дослідницька група розробила модель ПЗ для оптимізації розкладу університетських курсів. Їх модель ефективно врахувала численні обмеження, такі як наявність аудиторій, розклад викладачів та вимоги до курсів студентів. Впровадження моделі призвело до більш впорядкованого процесу складання розкладу, зменшення кількості конфліктів і підвищення рівня задоволеності студентів.

У дослідженні [18] було розроблено алгоритм, спеціально пристосований для оптимізації планування проєктів. Цей алгоритм використовував унікальне представлення для кодування різних аспектів плану проєкту, включаючи призначення завдань, терміни та розподіл ресурсів.

Також цей метод був предметом дослідження щодо його застосування в логістиці. Дослідження було присвячене використанню пошуку за допомогою табу для оптимізації маршрутів транспортних засобів для дистриб'юторської

компанії [19]. Алгоритм продемонстрував неабияку ефективність у скороченні відстаней і витрат, а також покращенні часу доставки.

Алгоритм імітації відпалу також є одним з поширених та потужних методів для оптимізації завдяки своєму застосуванню у плануванні виробництва. Дослідники застосували цей метод для оптимізації виробничих графіків на автомобільному заводі [20]. Алгоритм адаптувався до динамічного виробничого середовища, ефективно балансує між ефективністю виробництва та мінімізацією рівня відходів.

Поява методів машинного навчання, зокрема навчання з підкріпленням, відкрила нові можливості для оптимізації планування [21-23]. В одному з досліджень було застосовано навчання з підкріпленням для оптимізації управління запасами в контексті роздрібної торгівлі.

Алгоритм навчався на основі даних про продажі прогнозувати потреби в запасах, динамічно коригуючи замовлення, щоб мінімізувати дефіцит і надлишок запасів. Такий підхід не лише підвищив ефективність управління запасами, але й дав змогу зрозуміти, як споживачі купують товари [24].

Кожен з цих методів являє собою окремий підхід до вирішення проблем оптимізації планування.

Генетичні алгоритми почали використовуватись в галузі оптимізації завдяки унікальному підходу та універсальності [25]. Цей метод, працює, імітуючи процес природного відбору, де найбільш придатні рішення відбираються для рекомбінації та мутації, щоб еволюціонувати в кращі рішення протягом багатьох поколінь.

Прикладом застосування ГА є оптимізація складних мережевих конструкцій. Дослідники успішно використовували ГА для розробки оптимальних топологій мереж, демонструючи їхню здатність справлятися зі складними обмеженнями та множинними цілями [26]. Адаптивність ГА була ключовим фактором у досягненні рішень, які збалансували вартість, ефективність і надійність при проектуванні мереж.

ГА також мають певні недоліки. Однією з головних проблем є інтенсивність обчислень, особливо для великомасштабних задач. Процес розробки рішень для кількох поколінь може зайняти багато часу і вимагати значних обчислювальних ресурсів [27].

Крім того, існує ризик передчасної збіжності, коли алгоритм може зупинитися на неоптимальних рішеннях. Ця проблема вимагає ретельного налаштування параметрів, таких як швидкість мутації та розмір популяції.

1.3 Відомі засоби для оптимізації процесу планування проєктів

Задля досягнення мети необхідно провести аналіз відомих засоби та методи оптимізації.

Дослідження програмних і апаратних засобів для оптимізації планування проєктів і формування команд можна розділити на дві основні категорії: комерційні рішення і рішення з відкритим вихідним кодом.

Комерційні інструменти в цій галузі зазвичай пропонують розширені можливості та надійність, пристосовані до складних сценаріїв управління проєктами. Однак, вони часто пов'язані зі значними витратами і потребують кваліфікованого персоналу для ефективного використання [28].

Комерційні програми в цій галузі часто використовують складні фреймворки, пропонуючи надійні рішення, пристосовані до складних завдань планування проєктів і формування команд [29].

Ці рішення, як правило [30-31], більш відшліфовані і супроводжуються спеціальною підтримкою, але вони можуть бути дорогими і можуть вимагати спеціальних знань для ефективного впровадження.

До закритих програмних засобів для оптимізації команд та планування відносяться такі як наведені у таблиці 1.1.

Таблиця 2.1 – Засоби для оптимізації команд та планування

Назва компанії/проекту	Опис
Asana	Веб та мобільний додаток, розроблений, щоб допомогти командам організувати, відстежувати та керувати своєю роботою.
Trello	Інструмент для спільної роботи, який організовує проекти у вигляді дошок, списків та карток для кращого управління завданнями.
Monday.com	Хмарна платформа, яка дозволяє компаніям створювати власні додатки та інструменти для управління роботою.
Smartsheet	Корпоративна платформа для управління роботою та автоматизації, що використовується для оптимізації планування проектів.
Atlassian (Jira)	Відомий завдяки Jira, інструменту для відстеження проблем та управління проектами, особливо в розробці програмного забезпечення.
Workfront (Adobe)	Інструмент управління роботою для маркетологів, що пропонує налаштовані робочі процеси та інструменти стратегічного планування.

Продовження таблиці 2.1 - Засоби для оптимізації команд та планування

Назва компанії/проєкту	Опис
Wrike	Універсальна платформа для управління роботою, яка допомагає впорядкувати робочі процеси та підвищити ефективність проєктів.
ClickUp	Додаток, що поєднує в собі функції управління завданнями, постановки цілей і планування проєктів в одній платформі.

У сфері планування проєктів і формування команд платформи з відкритим кодом часто виявляються кращим вибором, і це явище має кілька різних, але взаємопов'язаних характеристик. Основною перевагою інструментів з відкритим вихідним кодом є їхня внутрішня кастомізація: ці платформи дозволяють здійснювати широкі модифікації відповідно до конкретних вимог і робочого процесу певної команди або проєкту [32].

Така гнучкість стає можливою завдяки доступності вихідного коду, що дозволяє компаніям пристосовувати програмне забезпечення до своїх конкретних потреб. З економічної точки зору, інструменти з відкритим кодом часто є економічно вигідним рішенням. Більшість таких інструментів є або безкоштовними, або вимагають мінімальних витрат, тим самим спрощуючи доступ для підприємств різного масштабу, від стартапів до великих корпорацій. Цей аспект проявляється у великій кількості ресурсів, які команди можуть використовувати, починаючи від допомоги у вирішенні проблем і закінчуючи розробкою інноваційних функцій [33].

Крім того, прозорість, притаманна програмному забезпеченню з відкритим кодом, підвищує безпеку та довіру. Користувачі мають можливість самостійно

перевіряти код, переконуючись у відсутності прихованих вразливостей чи шкідливих компонентів [34]. Нарешті, аспект інтеграції та сумісності є критично важливим фактором на користь інструментів з відкритим кодом. Найбільш популярні інструменти з відкритим кодом наведені у таблиці 2.2.

Таблиця 2.2 - Засоби для оптимізації команд та планування з відкритим кодом

Назва проєкту	Опис
GanttProject	Безкоштовний інструмент управління проєктами з відкритим вихідним кодом, що пропонує діаграми Ганта, управління ресурсами та календарі.
ProjectLibre	Альтернатива Microsoft Project з відкритим вихідним кодом, відома своєю сумісністю та можливостями планування проєктів.
OpenProject	Веб-система управління проєктами для співпраці команд, незалежно від їхнього місцезнаходження.
Redmine	Гнучкий веб-додаток для управління проєктами з функціями відстеження проблем, діаграмами Ганта та хронометражем.
Odoo	Набір бізнес-додатків з відкритим вихідним кодом, включаючи управління проєктами та CRM-функції.
MyCollab	Інструмент для управління проєктами та CRM, орієнтований на комплексне відстеження проєктів та співпрацю.
Kanboard	Простий інструмент управління проєктами з відкритим кодом, який використовує дошки Kanban для управління завданнями та проєктами.
LibrePlan	Інструмент для спільної роботи для планування, моніторингу та контролю проєктів, з відкритим вихідним кодом.

1.3.1 GanttProject

GanttProject - це інструмент управління проектами з відкритим вихідним кодом, який широко відомий своїм зручним інтерфейсом і потужною функціональністю, особливо у сфері планування та візуалізації термінів проекту. Його основні функції включають діаграми Ганта, управління ресурсами та управління завданнями, що робить його підходящим кандидатом для проектів, де чітке візуальне представлення та відстеження термінів є першочерговими.

Центральним елементом GanttProject є функціональність діаграм Ганта. Ці діаграми забезпечують візуальну шкалу часу для проекту, відображаючи завдання, їх тривалість, залежності та етапи, рисунок 1.1. Ця функція допомагає зрозуміти послідовність завдань та їхні критичні шляхи, що є важливим аспектом для ефективного планування проекту [35].

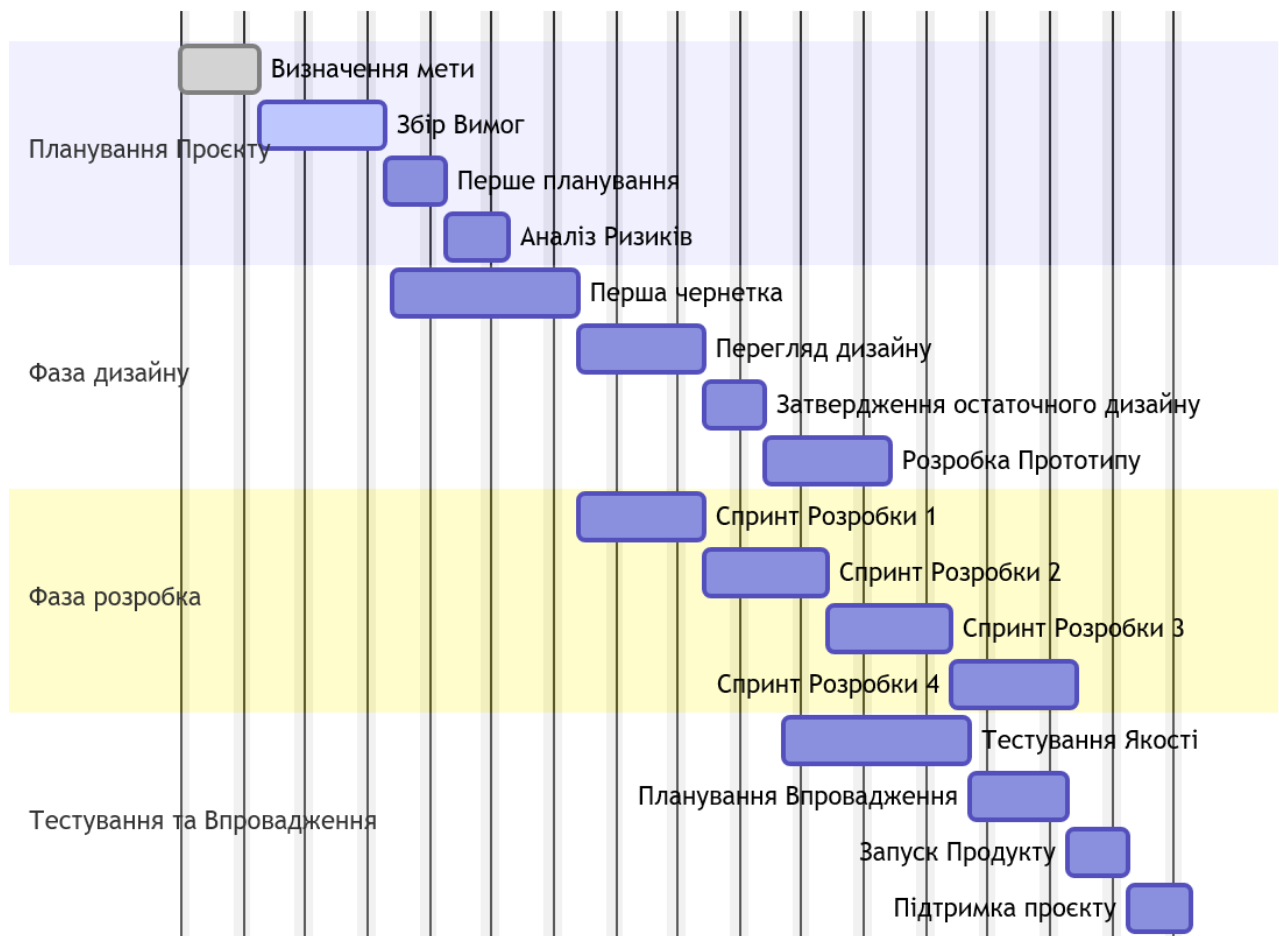


Рисунок 1.1 – Приклад діаграми Ганта

GanttProject дозволяє розподіляти та відстежувати ресурси. Це включає в себе призначення членів команди на завдання з урахуванням їх доступності та завантаженості. Ця функція життєво важлива для збалансування розподілу ресурсів і забезпечення того, щоб члени команди не були надмірно або недостатньо використані. Він підтримує створення структури розбиття проєкту [36], яка є ієрархічним представленням того, що має бути досягнуто в рамках проєкту. Ця функція допомагає в організації та ефективному управлінні обсягом проєкту.

Інструмент пропонує можливість експортувати дані в різні формати, такі як PDF і HTML, що дозволяє легко ділитися ними і створювати звіти. Ця функція має вирішальне значення для інформування зацікавлених сторін про хід виконання проєкту.

До переваг можна віднести інтуїтивно зрозумілі діаграми Ганта, що забезпечують чітке і стисле візуальне представлення часової шкали проєкту, що полегшує планування і відстеження, а також те, що функції управління ресурсами GanttProject дозволяють ефективно розподіляти членів команди, забезпечуючи збалансований розподіл робочого навантаження.

Проте для надзвичайно складних проєктів, особливо тих, що потребують розширених функцій, таких як управління ризиками або інтегровані інструменти командної комунікації, GanttProject може бути недостатньо ефективним. Хоча GanttProject ефективний для планування, йому не вистачає розширених функцій співпраці, таких як оновлення в реальному часі або інтегровані комунікаційні платформи, які є важливими у великих, динамічних командних середовищах.

1.3.2 ProjectLibre

ProjectLibre - це відоме програмне забезпечення для управління проєктами з відкритим вихідним кодом, яке часто хвалять за його комплексні функції, що задовольняють широкий спектр потреб в управлінні проєктами. Він є помітною альтернативою Microsoft Project, пропонуючи подібні функціональні можливості

без пов'язаних з ними витрат, що робить його доступним для широкого кола користувачів, від індивідуальних менеджерів проєктів до великих організацій.

В основі можливостей ProjectLibre лежить потужний набір функцій планування, який включає в себе розширені інструменти діаграми Ганта, механізми розподілу ресурсів і мережеві діаграми. Функціональність діаграми Ганта, зокрема, є особливою особливістю, що дозволяє детально візуалізувати часові рамки проєкту і залежності між завданнями, рисунок 1.2.

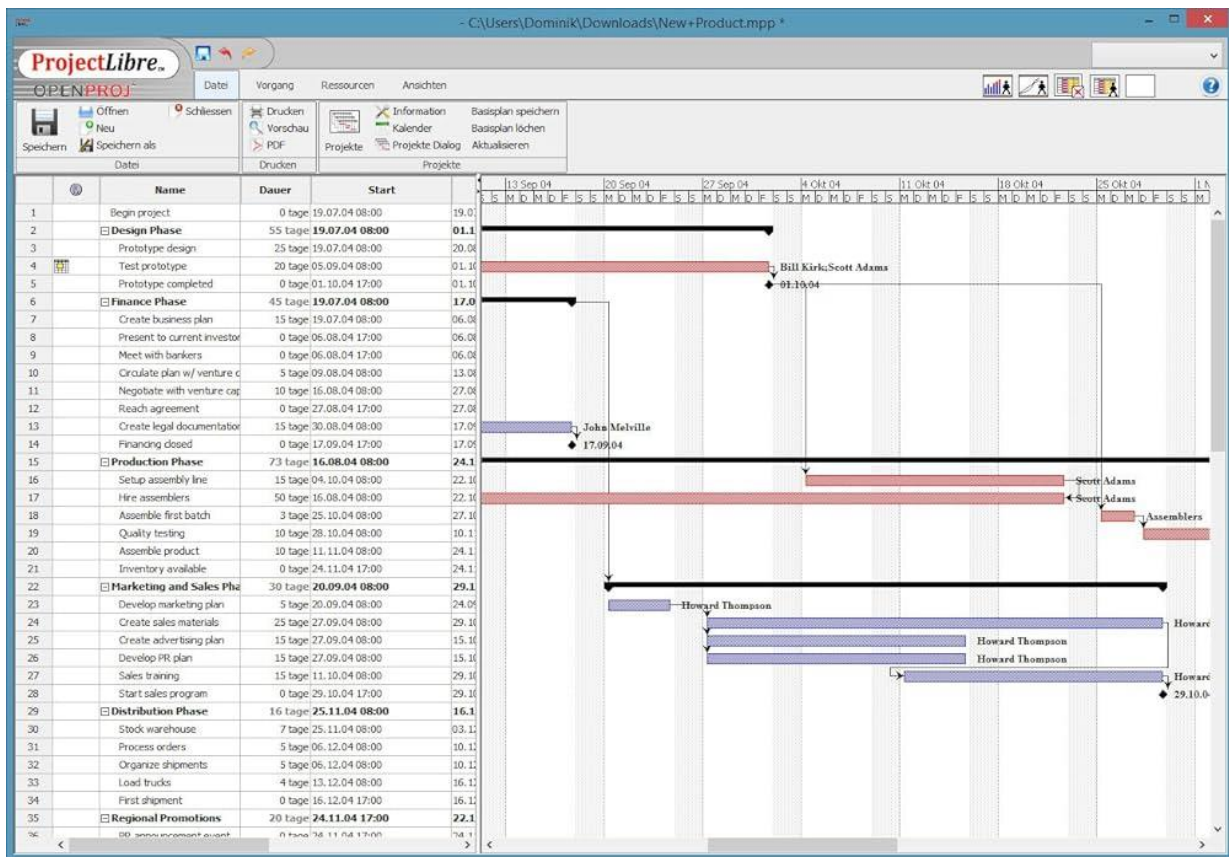


Рисунок 1.2 – Графічний інтерфейс програмного забезпечення ProjectLibre [37]

Цей аспект ProjectLibre допомагає визначити послідовність дій і виявити критичні шляхи, що є важливим для ефективного планування [38] та управління проєктами.

З точки зору управління ресурсами, ProjectLibre вирізняється своєю здатністю керувати різними ресурсами, чи то людськими, чи то матеріальними.

Інструмент дозволяє розподіляти ресурси на конкретні завдання і відстежувати їх використання протягом усього життєвого циклу проєкту.

Ця функціональність є особливо цінною для оптимізації формування команди і забезпечення ефективного використання ресурсів, уникаючи таким чином вузьких місць і перерозподілу ресурсів.

Ще одним важливим аспектом ProjectLibre є можливості мережевих діаграм, які надають альтернативний погляд на завдання проєкту та їх взаємозалежності.

Ці діаграми пропонують більш орієнтовану на процес перспективу проєкту, доповнюючи вигляд діаграми Ганта і покращуючи загальне розуміння потоку і структури проєкту.

Однак, хоча ProjectLibre пропонує потужний набір інструментів для планування та управління проєктами, він не позбавлений недоліків. Однією з помітних проблем є його користувацький інтерфейс, який, хоча і є функціональним, може бути не таким інтуїтивно зрозумілим, як деякі з більш сучасних інструментів управління проєктами.

Це потенційно може призвести до того, що новим користувачам доведеться довше вчитися [39]. Крім того, хоча ProjectLibre ефективно полегшує планування проєктів, йому може бракувати деяких з більш просунутих функцій, які можна знайти у високоякісному програмному забезпеченні для управління проєктами, таких як поглиблений аналіз ризиків або інтегровані платформи для командної співпраці.

У контексті проєкту, що зосереджений на оптимізації планування проєкту та формуванні команди, ProjectLibre є потужним інструментом, особливо для тих, хто шукає недороге, але всеохоплююче рішення. Його сильні сторони в плануванні, управлінні ресурсами та візуалізації проєкту добре узгоджуються з потребами такого проєкту.

Однак, для проєктів, які потребують більш просунутих функцій, особливо з точки зору співпраці та комунікації у реальному часі, може знадобитися вийти

за межі ProjectLibre або інтегрувати його з іншими інструментами, щоб досягти бажаного результату

1.3.3 OpenProject

OpenProject - це відоме програмне забезпечення для управління проектами з відкритим вихідним кодом, яке вирізняється широким набором функцій, орієнтованих на спільне управління проектами. Його дизайн розрахований на широкий спектр типів і розмірів проектів, що робить його особливо універсальним у сфері планування проектів і координації роботи команди.

Однією з головних переваг OpenProject є його можливості для спільної роботи. Платформа побудована з орієнтацією на командну роботу в основі, пропонуючи такі функції, як спільні дошки завдань, форуми та вікі.

На додаток до інструментів для спільної роботи, OpenProject також пропонує надійні засоби планування проектів, включаючи розширені діаграми Ганта та дорожні карти (рисунок 1.3).

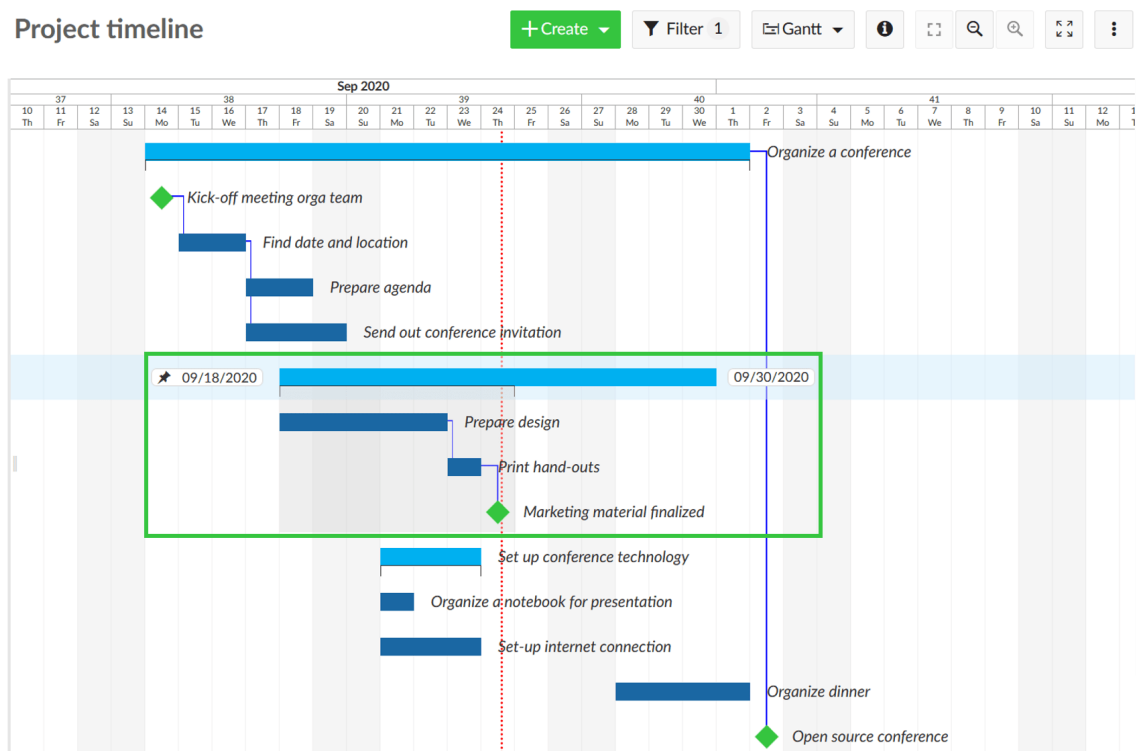


Рисунок 1.3 – Інтерфейс програмного забезпечення OpenProject [40]

Ці інструменти дозволяють детально планувати і візуалізувати часові рамки і етапи проєкту. Функціональність діаграми Ганта в OpenProject особливо примітна своєю інтерактивною природою, що дозволяє менеджерам проєктів легко створювати, змінювати і пов'язувати завдання безпосередньо в межах діаграми. Такий рівень інтерактивності підвищує легкість планування та перепланування завдань, що є життєво важливим у динамічному проєктному середовищі, де часто відбуваються зміни.

Сильні сторони OpenProject у плануванні проєктів ще більше посилюються завдяки функціям відстеження часу та звітування про витрати. Ці функції дозволяють ретельно відстежувати час, витрачений на різні завдання, і пов'язані з ними витрати, що є важливим для ефективного управління бюджетом проєкту. Цей аспект має вирішальне значення в проєктах, де розподіл ресурсів і дотримання бюджету є ключовими факторами. Здатність програмного забезпечення інтегрувати дані про час і витрати у звітність за проєктом забезпечує всебічний огляд стану проєкту, що дозволяє приймати обґрунтовані рішення.

Що стосується основних математичних моделей або формул, OpenProject, як і багато інших інструментів управління проєктами, ймовірно, використовує ряд алгоритмів для різних функціональних можливостей, таких як планування та оптимізація ресурсів. Наприклад, при вирівнюванні та розподілі ресурсів алгоритми зазвичай збалансовують розподіл ресурсів протягом тривалості проєкту:

$$\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \quad (1.1)$$

Потенційно використовуючи моделі лінійного програмування або евристичні підходи для оптимізації використання ресурсів з урахуванням таких обмежень, як час [41-43], вартість і доступність:

$$\sum_{j=1}^m x_{ij} \leq R_i, \forall i \quad (1.2)$$

$$\sum_{i=1}^n x_{ij} \leq T_j, \forall j, \text{ де } x_{ij} \geq 0 \quad (1.3)$$

У цьому представленні c_{ij} – це вартість, пов'язана з виділенням ресурсу i на завдання j , x_{ij} – кількість ресурсу i , виділеного на завдання j , R_i - загальна доступна кількість ресурсу i , а T_j - загальна потреба в ресурсі для виконання завдання j . Цей тип оптимізаційної задачі, хоча і є спрощеною моделлю, є типовим у сценаріях розподілу ресурсів і є частиною алгоритмів, що використовуються у програмному забезпеченні OpenProject.

Потужний набір інструментів управління проектами OpenProject у поєднанні з його сильним акцентом на співпраці та адаптивності робить його універсальним вибором для управління складними проектами, які вимагають динамічного планування та ефективної командної співпраці. Хоча OpenProject не спеціалізується виключно на оптимізації ресурсів або алгоритмах планування, його збалансований підхід до управління проектами робить його універсальним інструментом в арсеналі планувальника проектів, особливо в сценаріях, де гнучкість, співпраця і широкий нагляд за проектом є ключовими.

1.4 Постановка задачі дослідження

Аналіз наявних інструментів та методів для планування та управління роботою команд розробників показав, що існуючі підходи не є достатньо ефективними для повноцінного автоматизованого планування. Це вказує на необхідність розробки нових методів та інструментів в рамках інформаційної технології для покращення інтелектуалізації планувальника роботи команд розробників. Ключові кроки для досягнення цієї мети включають:

1. Дослідити методи та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.
2. Дослідити проблему оптимізації планування.
3. Проаналізувати відомі засоби та методи оптимізації.
4. Обґрунтувати вибір теоретичних та експериментальних методів для дослідження.
5. Підготувати дані для обробки генетичним алгоритмом.
6. Створити структуру інформаційної технології
7. Розробити удосконалений метод інтелектуалізованого планувальника роботи команди розробників проєктів
8. Провести експериментальне дослідження методу
9. Розробити архітектуру інтелектуалізованого планувальника роботи команди розробників проєктів
10. Скласти вимоги до інтелектуалізованого планувальника роботи команди розробників проєктів
11. Провести оцінку ефективності розробленого алгоритму.

2 ОБҐРУНТУВАННЯ І ВИБІР ТЕОРЕТИЧНИХ ТА ЕКСПЕРИМЕНТАЛЬНИХ МЕТОДІВ ДОСЛІДЖЕННЯ

2.1 Теоретичні основи генетичних алгоритмів у плануванні проєктів та формуванні команд

Для досягнення мети необхідно обґрунтувати вибір теоретичних та експериментальних методів для дослідження.

В основі методу лежить використання ГА - класу еволюційних алгоритмів, які імітують процес природного відбору. Генетичні алгоритми особливо потужні у вирішенні оптимізаційних задач завдяки своїй здатності ефективно шукати рішення у великому просторі рішень. У контексті планування проєктів і формування команд ГА пропонують надійну основу для визначення оптимальних комбінацій ресурсів, навичок і термінів, подібно до процесу природної еволюції, де виживають і поширюються найбільш пристосовані рішення [43].

Обґрунтування вибору ГА базується на їхній продемонстрованій ефективності в роботі зі складними, багатовимірними просторами рішень, які є типовими для сценаріїв управління проєктами. На відміну від традиційних інструментів управління проєктами, які часто покладаються на лінійні, детерміновані підходи, ГА перебирають безліч потенційних рішень [44], розвиваючись і адаптуючись до оптимального або близького до оптимального рішення.

Алгоритм зазвичай починається з випадково згенерованої популяції потенційних рішень, кожне з яких представлене хромосомоподібною структурою даних. У контексті управління проєктами хромосома може представляти конкретний план проєкту [45] з різними параметрами, такими як розподіл завдань, ресурсів і термінів.

Потім ГА використовує функцію пристосованості для оцінки кожної хромосоми. Ця функція є критично важливою, оскільки вона визначає, наскільки

"придатним" або відповідним є рішення. Для планування проєкту фітнес-функція може оцінювати такі фактори, як ефективність використання ресурсів, дотримання термінів або баланс робочого навантаження команди [46]. Функція придатності може бути представлена у вигляді:

$$F = \alpha \times E + \beta \times D + \gamma \times B, \quad (2.1)$$

де α , β і γ – ваги, що відображають важливість кожного фактора, E - ефективність розв'язку, яка оцінює якість та результативність вирішення задачі або проблеми, D - дотримання дедлайнів розв'язку, що відображає здатність розв'язку виконуватися в рамках встановлених термінів [47], B – збалансованість навантаження розв'язку, що показує, наскільки рівномірно ресурси та зусилля розподіляються у процесі виконання розв'язку.

ГА включає такі оператори, як відбір, де рішення з вищими показниками пристосованості з більшою ймовірністю будуть обрані для розмноження. Кросинговер і мутації вносять мінливість і нові ознаки в популяцію, аналогічно до біологічної еволюції. Кросинговер можна уявити як обмін сегментами між двома батьківськими хромосомами, тоді як мутація передбачає випадкові зміни в хромосомі.

Таким чином, теоретична основа для використання ГА у плануванні проєктів та формуванні команд ґрунтується на їхній здатності досліджувати та використовувати складний простір рішень, ітеративно еволюціонуючи у напрямку до найбільш підходящих рішень [48].

Для того, щоб перевірити ефективність запропонованого підходу на основі ГА, було прийнято подвійний експериментальний дизайн.

По-перше, було створено контрольоване середовище моделювання, що дозволяє маніпулювати змінними та спостерігати за результатами у повторюваний [49], послідовний спосіб. Це моделювання передбачало створення віртуальних сценаріїв проєкту з різним рівнем складності, ресурсних обмежень та командної динаміки.

Паралельно з цим було проведено реальне застосування моделі на основі ГА. Це передбачало впровадження алгоритму в існуючу систему управління проектами в корпоративному середовищі.

Ця двостороння стратегія забезпечує комплексну оцінку ефективності моделі на основі ГА [50]. Керована симуляція слугує лабораторією для перевірки гіпотез і спостереження за поведінкою ГА в різних контрольованих умовах. Сценарії віртуальних проектів, розроблені з різними рівнями складності, ресурсними обмеженнями та командною взаємодією, слугують тестовим майданчиком. Ці сценарії дозволяють систематично аналізувати роботу ГА, виявляючи його сильні та слабкі сторони в управлінні проектами.

У той же час, застосування моделі на основі ГА в реальному бізнес-середовищі дає безцінну інформацію про її практичну корисність і адаптивність. На цьому етапі алгоритм ГА інтегрується в існуючу систему управління проектами для перевірки його сумісності та ефективності в реальних умовах. Етап реального застосування є критично важливим, оскільки він надає емпіричні докази застосовності моделі та її впливу в реалістичному середовищі. Цей етап також сприяє виявленню непередбачуваних викликів і можливостей, які можуть виникнути при переході від теоретичної моделі до практичного інструменту в динамічному бізнес-контексті.

Комплексний аналіз моделі на основі ГА за допомогою цих двох методологій має вирішальне значення. Контрольоване моделювання дозволяє отримати детальне, точне розуміння теоретичних основ моделі та її поведінки в контрольованих умовах. На відміну від цього, реальне застосування оцінює її практичність, адаптивність і загальний вплив в реальному, операційному середовищі. Такий подвійний підхід забезпечує цілісну оцінку, яка враховує як теоретичну надійність, так і практичну життєздатність, тим самим підтверджуючи ефективність моделі в поліпшенні процесів планування проектів і формування команд.

Цей подвійний підхід: контрольоване моделювання та реальне застосування, це показано на рисунку 2.1.

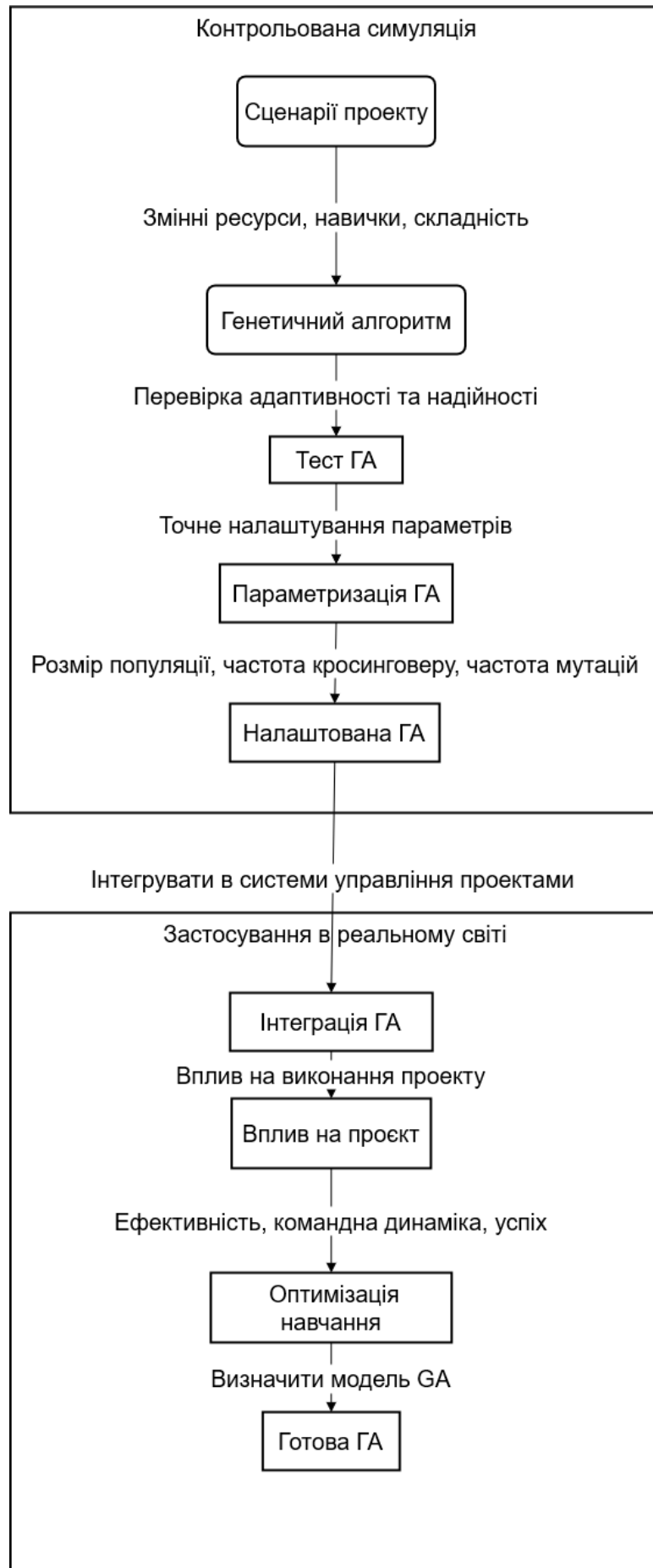


Рисунок 2.1 – Діаграма подвійного підходу до дизайну системи

Аспект симуляції дозволяє поглиблено дослідити поведінку ГА за різних контрольованих умов. Це передбачає створення декількох сценаріїв проєкту з різними рівнями ресурсних обмежень, різноманітністю навичок команди та складністю проєкту [52-54]. Змінюючи ці змінні, можна ретельно перевірити адаптивність і надійність ГА.

Мета полягала у спостереженні за практичними викликами та перевагами застосування ГА в реальному середовищі, надаючи важливу інформацію про ефективність алгоритму в реальних сценаріях.

Моделювання також полегшує точне налаштування параметрів ГА, таких як розмір популяції, частота кросинговеру і частота мутацій, щоб знайти найефективнішу комбінацію для планування проєкту [55]. Ці параметри можна виразити так:

- розмір популяції (N): кількість рішень у кожному поколінні;
- частота кросинговеру (r_c): ймовірність кросинговеру між парами хромосом;
- частота мутацій (r_m): ймовірність мутації в хромосомі.

Паралельно з цим, реальне застосування моделі на основі ГА в організаційному середовищі дозволяє отримати практичні знання. Цей етап передбачає інтеграцію ГА в існуючі системи управління проєктами і спостереження за його впливом на ефективність виконання проєкту, динаміку команди і загальний успіх проєкту.

Виклики, що виникають на цьому етапі, такі як неузгодженість даних або проблеми системної інтеграції, надають цінні можливості для навчання та вдосконалення моделі ГА.

2.2 Підготовка даних та обробка генетичним алгоритмом для оптимального планування проєкту

Для досягнення мети необхідно підготувати дані для обробки генетичним алгоритмом.

Перед розгортанням ГА дані повинні бути ретельно попередньо оброблені. Це передбачає очищення, нормалізацію та кодування даних у формат, придатний для обробки ГА [56].

Нормалізація, наприклад, має вирішальне значення для забезпечення того, щоб усі параметри оцінювалися за порівнянною шкалою, особливо коли маємо справу з різними типами даних, такими як тривалість часу, кошторис витрат і рівень кваліфікації.

Нормалізацію можна представити так:

$$NV = \frac{FV - \text{Min}V}{\text{Max}V - \text{Min}V}, \quad (2.2)$$

де NV відображає нормалізоване значення, що є результатом віднімання мінімального значення (MinV) з фактичного значення (MaxV) та подальшого ділення цієї різниці на різницю між максимальним (MaxV) та мінімальним значеннями, що забезпечує масштабування фактичного значення у визначений діапазон, тим самим уніфікуючи його для подальшого аналізу або порівняння [57].

Окрім базової нормалізації, важливу роль у підвищенні продуктивності ГА відіграє функціональна інженерія. Вона передбачає створення нових вхідних функцій на основі наявних даних, що потенційно може виявити приховані закономірності, важливі для планування проекту [58].

Наприклад, отримання таких показників, як "складність завдання" або "синергія команди", на основі вихідних даних, таких як тривалість завдання і навички окремих членів команди, може забезпечити більш глибокі вхідні дані для ГА.

Потім ГА обробляє ці дані за допомогою серії еволюційних кроків. Відбір відбувається на основі фітнес-функції, яка забезпечує виживання і розмноження найбільш пристосованих рішень.

Кросинговер і мутації вносять нові характеристики в популяцію, збільшуючи різноманітність і потенціал рішень. Процес відбору можна змоделювати як імовірнісну функцію, де рішення з вищою пристосованістю мають більшу ймовірність бути обраними для наступного покоління.

Враховуючи різноманітність типів проектів, сегментація та стратифікація даних гарантує, що ГА не буде упередженим до будь-якої конкретної категорії проектів [59].

Сегментуючи дані на основі характеристик проекту, таких як розмір, складність або сфера, ГА можна налаштувати так, щоб він однаково добре справлявся з різними сценаріями.

Етап обробки завершується створенням оптимізованих проектних планів і конфігурацій команд, які потім оцінюються за реальними критеріями здійсненності та ефективності.

Планування проекту часто передбачає збалансування декількох цілей, таких як мінімізація витрат при максимізації якості або дотримання строгих термінів. Впровадження багатоцільового ГА дозволяє оптимізувати ці конфліктуючі цілі одночасно [60]. Це можна представити у вигляді фронту Парето, де кожна точка представляє рішення, яке не домінує над будь-яким іншим у всіх цілях.

Щоб оптимізувати роботу ГА в реальному середовищі, також можна інтегрувати автоматизовані системи подачі та вилучення даних. Це передбачає налаштування інтерфейсів з існуючими інструментами управління проектами для автоматичного вилучення відповідних даних (завдання, терміни, ресурси) і завантаження результатів роботи ГА (оптимізовані розклади і конфігурації команд) назад в систему.

Інформаційна технологія, характеризується циклічним потоком даних. Спочатку в систему вводяться параметри проекту, такі як завдання, терміни і пули ресурсів. Потім ці вхідні дані обробляються ГА [61], який ітеративно шукає найефективніший план проекту та склад команди, ітеративна структура представлена на рисунку 2.2.

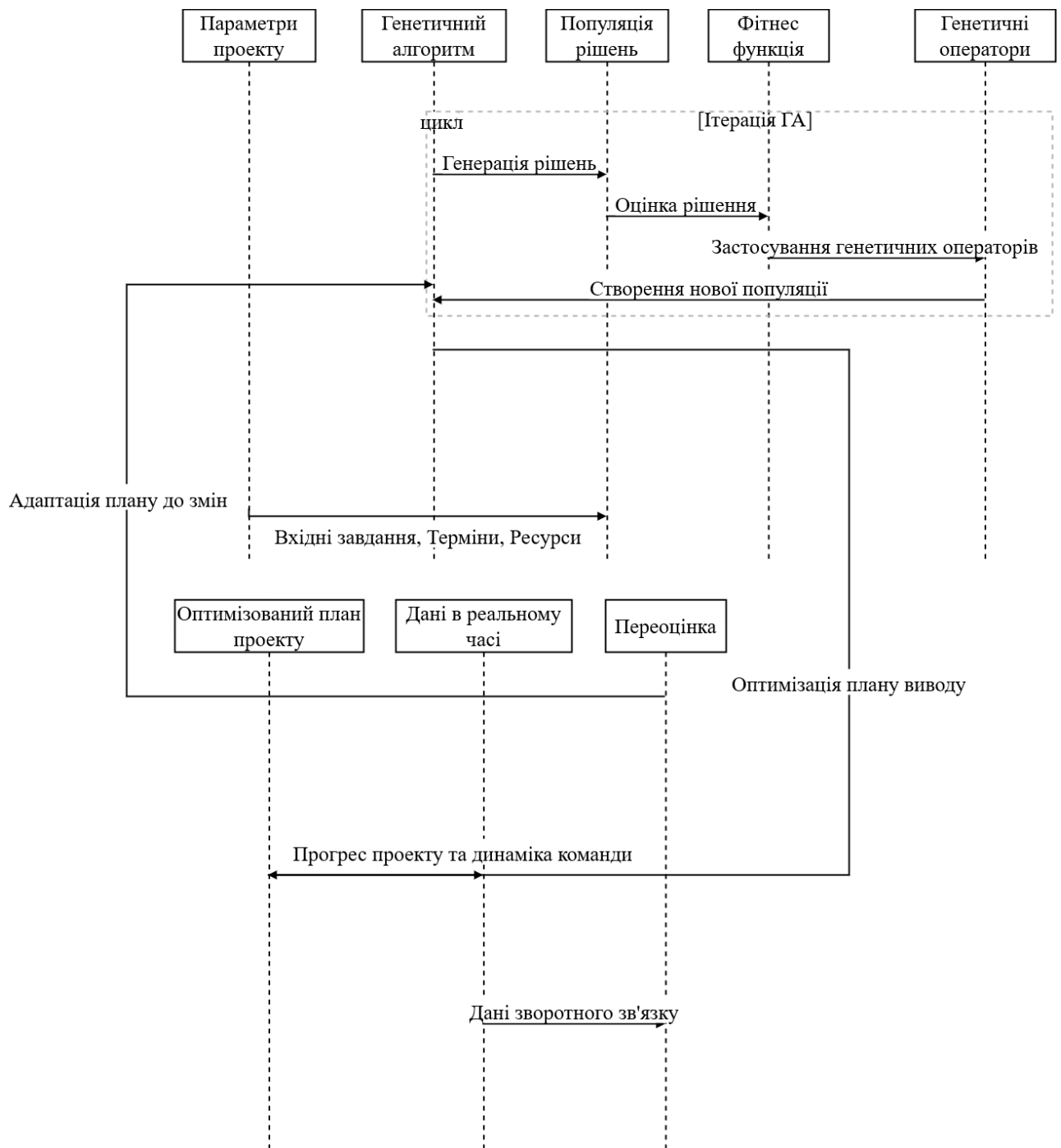


Рисунок 2.2 Схема ітеративної структури методу

Кожна ітерація ГА передбачає генерування популяції рішень, оцінку цих рішень за допомогою функції пристосованості, а потім застосування генетичних операторів для створення нової популяції [62]. Цей цикл повторюється до тих пір, поки не буде знайдено задовільне рішення або не буде досягнуто заданої кількості поколінь.

Інформаційний потік у розробленому методі характеризується безперервним циклом введення, обробки та виведення даних. На етапі введення відбувається збір даних, пов'язаних з проектом, включаючи завдання, дедлайни, наявність ресурсів та профілі членів команди. Потім ці дані подаються в ГА, де вони проходять обробку - селекцію, кросинговер, мутацію та оцінку придатності.

2.3 Структура інформаційної системи

Для досягнення мети необхідно створити структуру інформаційної технології.

При розробці інформаційної системи для покращення планування проектів та формування команд за допомогою генетичного алгоритму, підхід поєднує в собі кілька складних елементів [63-65]. Ця система, призначена для покращення виконання та управління проектами, використовує генетичні алгоритми для оптимізації. Її архітектура включає окремі, але взаємопов'язані частини: клієнтський і серверний компоненти, базу даних та прикладний програмний інтерфейс (API).

В основі клієнтського компоненту лежить зручний інтерфейс, розроблений таким чином, щоб бути інтуїтивно зрозумілим та інформативним. Цей інтерфейс є основною точкою взаємодії для менеджерів проектів, членів команди та інших зацікавлених сторін. Клієнтська частина надає кілька ключових функціональних можливостей.

По-перше, він дозволяє вводити і відстежувати деталі проекту, такі як обсяг, графік, ресурси і специфічні вимоги. Ця функція має вирішальне значення для поточного нагляду за проектом.

По-друге, система надає рекомендації щодо складу команди, отримані на основі генетичного алгоритму. Ці пропозиції враховують кілька факторів, включаючи навички, доступність та попередні результати роботи потенційних членів команди. Нарешті, клієнтський компонент допомагає збирати відгуки про роботу команди та прогрес проекту.

Цей зворотній зв'язок є важливим для вдосконалення алгоритму та забезпечення постійного вдосконалення управління проектами та формування команди [66].

Серверний компонент цієї інформаційної системи відіграє ключову роль у загальній архітектурі. Як основа системи, на нього покладені критичні функції обробки та зберігання даних. Можливості сервера відіграють центральну роль в ефективності та результативності системи [67], обробляючи величезні обсяги даних і забезпечуючи надійне зберігання інформації та швидкий доступ до неї в разі потреби.

Надійне проектування та реалізація цього сегменту має вирішальне значення для безперебійної роботи та надійності всієї інформаційної системи.

Інтеграція цих компонентів формує цілісну систему, кожна частина якої доповнює інші для досягнення кінцевої мети - оптимізації планування проектів та формування команди [68]. Використання генетичного алгоритму в цій системі забезпечує рівень складності та адаптивності, що дозволяє приймати динамічні та ефективні рішення, пристосовані до конкретних потреб проекту та динаміки команди.

Ретельне узгодження цих елементів підкреслює потенціал системи докорінно змінити спосіб планування проектів і формування команд, що в кінцевому підсумку сприяє підвищенню ефективності та успіху в різних проектних починаннях. При побудові інформаційної системи з використанням генетичного алгоритму для планування проектів і формування команд інтегруються кілька ключових елементів, які формують ядро її функціональності.

Ця система підтримується низкою взаємопов'язаних компонентів, кожен з яких відіграє життєво важливу роль [69] у забезпеченні ефективної оптимізації складу команди та управління проектами.

В основі системи лежить генетичний алгоритм, де відбувається первинний процес оптимізації. Його завданням є дослідження безлічі потенційних командних комбінацій та оцінка кожної з них на основі набору попередньо

визначених показників успіху [70]. Ефективність цього механізму має вирішальне значення, оскільки він визначає загальну ефективність процесу формування команди, визначаючи найбільш підходящі конфігурації команд для кожного конкретного проекту.

Обробка даних та логічна обробка - ще один важливий аспект системи. Цей сегмент відповідає за управління потоком вхідних даних, виконання алгоритмічних розрахунків і забезпечення безперебійної передачі результатів у клієнтський інтерфейс. Невід'ємною частиною цього процесу є надійна та надійна база даних, спеціально розроблена для зберігання та управління різноманітними типами даних. Ця база даних є важливою частиною системи [71], що лежить в основі її функціональності та забезпечує цілісність даних.

Система також включає комплексне сховище проектних даних, включаючи такі деталі, як параметри проекту, цілі, терміни і потреби в ресурсах. Ці дані створюють необхідний контекст для ефективної роботи генетичного алгоритму, що дозволяє йому підбирати склад команди відповідно до конкретних потреб кожного проекту.

Крім того, існує велика база даних членів команди, яка містить детальні профілі потенційних членів команди. Ці профілі включають інформацію про навички, досвід, доступність та минулі досягнення, що дає повне уявлення про можливості кожної особи та її придатність для виконання різних проектних ролей. Історія та аналіз проекту також є важливим компонентом. Цей сегмент збирає та аналізує історичні дані про минулі проекти та результати роботи команди. Цей постійний збір даних допомагає вдосконалювати та підвищувати ефективність алгоритму з часом, що дозволяє постійно навчатися та вдосконалювати процес відбору команд. По суті, поєднання цих елементів в рамках системи створює складний і потужний інструмент для планування проектів і формування команд. Інтеграція генетичного алгоритму з добре структурованою системою обробки та зберігання даних забезпечує потужне рішення для оптимізації результатів проекту та ефективності роботи команди (рисунки 2.5).

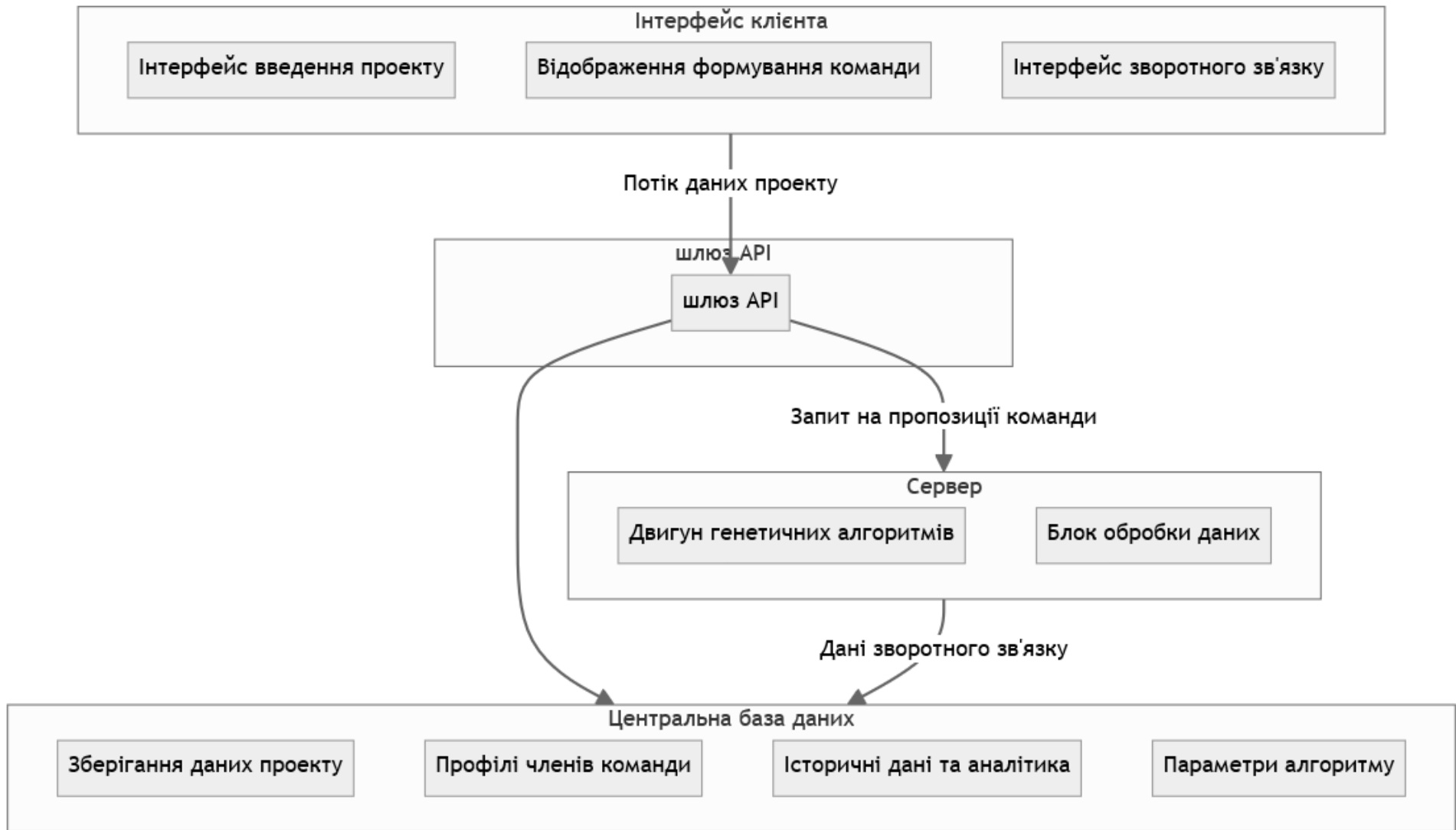


Рисунок 2.5 Архітектура системи управління проектами

Система також включає розділ, присвячений параметрам і моделям алгоритму. Ця область містить дані, пов'язані з самим генетичним алгоритмом, такі як критерії відбору, швидкість мутацій і функції пристосованості. Конфігурація та налаштування цих параметрів є ключовими для здатності алгоритму точно та ефективно визначати оптимальні конфігурації команд.

Нарешті(API) відіграє важливу роль як посередник, що забезпечує безперебійну комунікацію між клієнтськими та серверними компонентами. Він дозволяє передавати дані про проекти та команди з боку клієнта та отримувати оптимізовані конфігурації команд.

Крім того, API дозволяє інтегруватися з іншими інструментами управління проектами або HR-системами, підвищуючи корисність системи і точність оброблюваних даних.

Потік даних у цій інформаційній системі починається з введення даних від менеджерів проектів або автоматизованих каналів, що деталізують вимоги до проекту та наявних членів команди.

Ці дані обробляються сервером, де генетичний алгоритм оцінює численні формування команд, оптимізуючи їх для успіху проекту. Потім рекомендовані команди та проектні стратегії представляються користувачам через клієнтський інтерфейс.

Відгуки та дані про ефективність завершених проектів повертаються в систему, що дозволяє генетичному алгоритму навчатися і адаптуватися, покращуючи майбутні рекомендації (рисунок 2.4 та 2.5).

Система включає алгоритм динамічного планування, що має вирішальне значення для ефективного управління проектами. Цей алгоритм розподіляє ресурси та планує завдання відповідно до мінливих вимог проекту. Він враховує такі обмеження, як доступність ресурсів, залежність завдань і дедлайни. Алгоритм планування постійно відстежує прогрес проекту [72, 73] і може коригувати розклад у відповідь на несподівані зміни або затримки. Гнучкість має важливе значення для підтримання динаміки проекту та забезпечення своєчасного завершення.

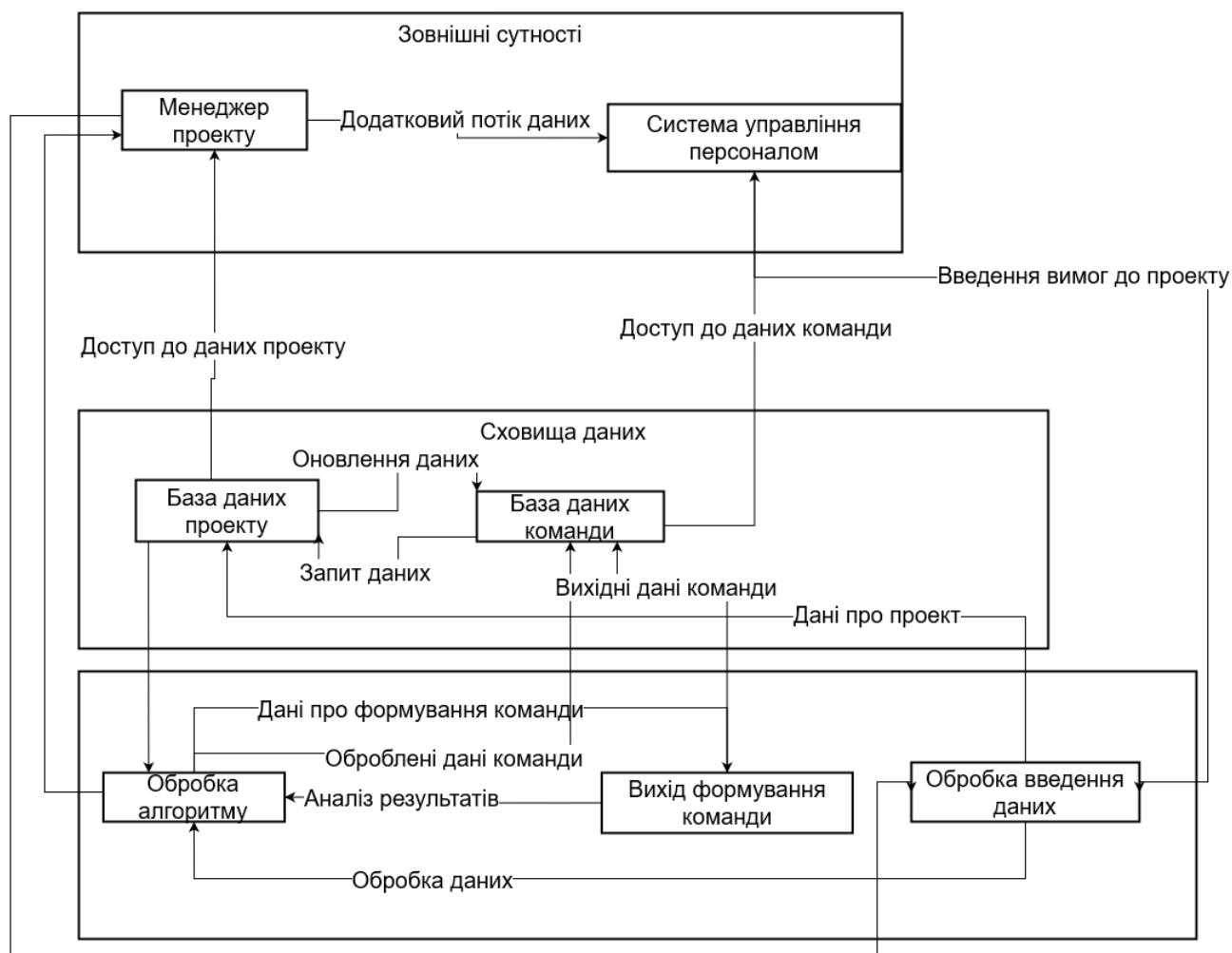


Рисунок 2.4 Потік даних у інформаційній системі

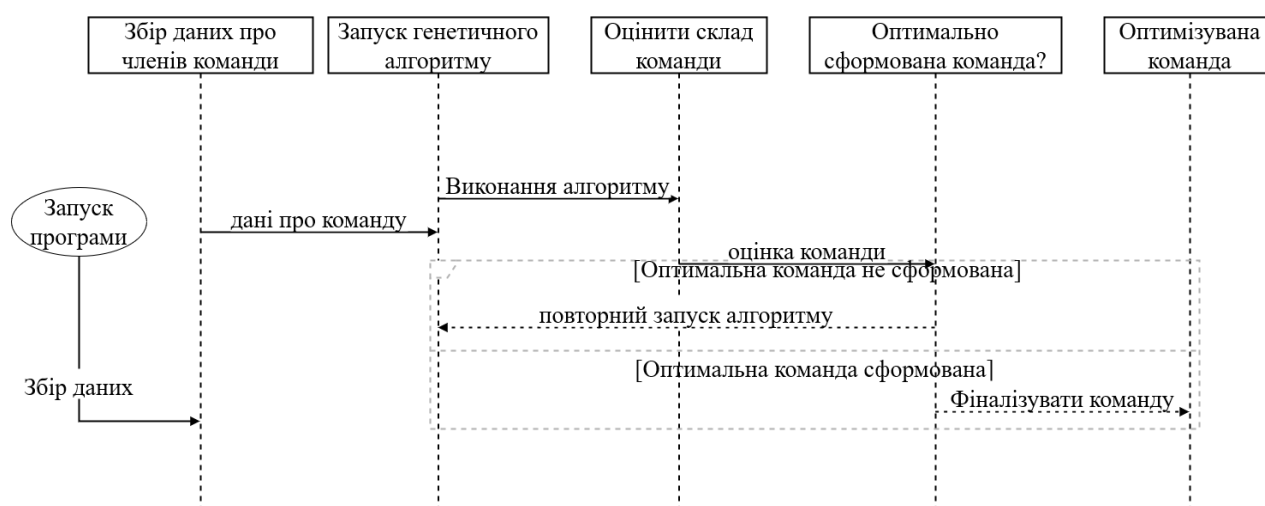


Рисунок 2.5 – Схема процесу оптимізації команди

Іншим важливим аспектом є використання методів машинного навчання для покращення процесу прийняття рішень генетичним алгоритмом. Аналізуючи

історичні дані про результати проєктів [74] та ефективність роботи команди, система може виявити закономірності та ідеї, які допоможуть сформувати більш ефективний склад команди та проєктні стратегії.

Алгоритми машинного навчання разом з іншими складовими дозволяють виконувати усі поставлені функції, на рисунку 2.6 показана схема класів яка була використана у розробці.

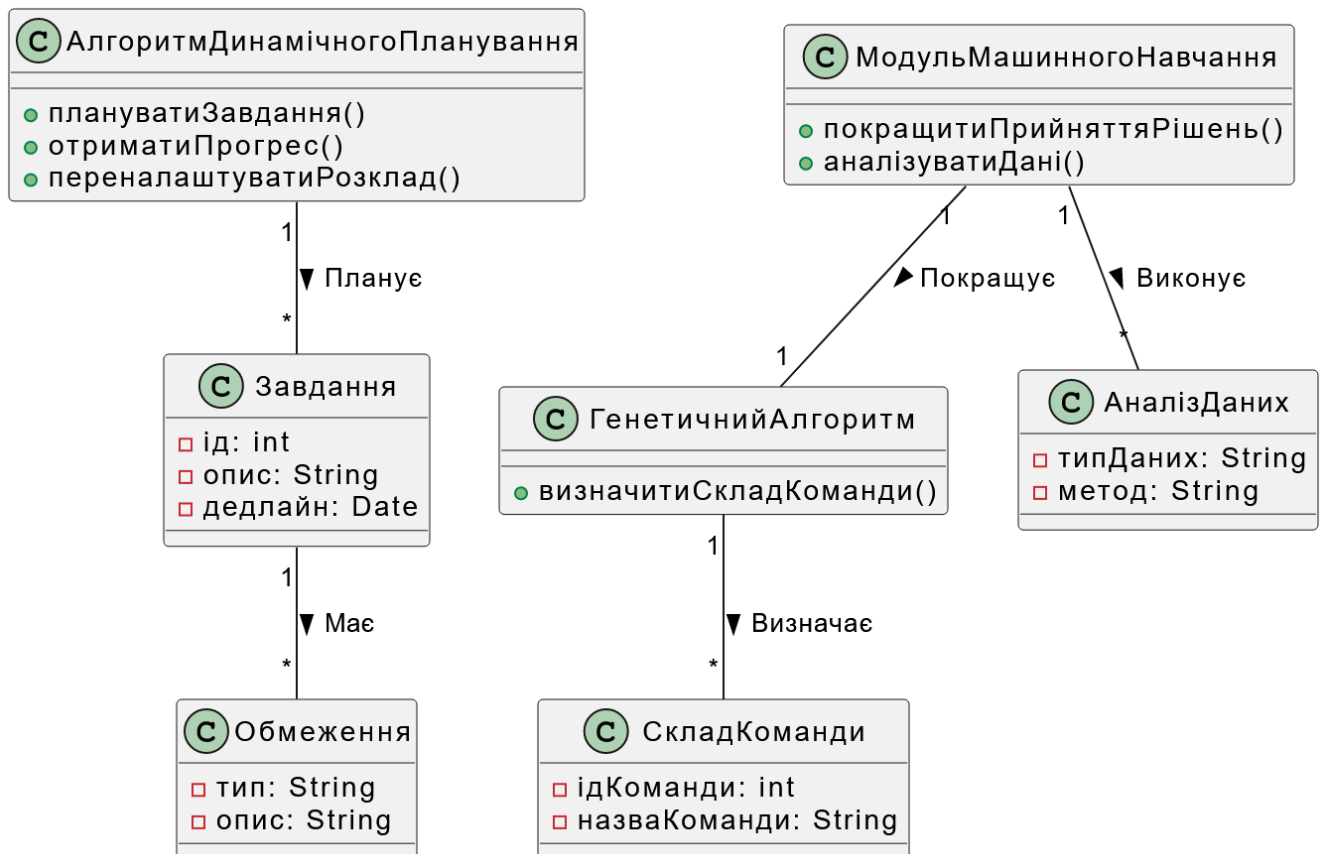


Рисунок 2.6 – Блок схема використаних класів

Інтерфейс системи також має сучасні інструменти візуалізації, які відображають складні дані та результати аналізу в зручному та зрозумілому для користувача вигляді. Інтерактивні дашборди пропонують метрики проєкту в режимі реального часу, статистику роботи команди та алгоритмічні результати.

Ця функція має важливе значення для прийняття обґрунтованих рішень та сприяння прозорості серед зацікавлених сторін [75]. Система використовує

найсучасніші механізми шифрування та контролю доступу для забезпечення безпеки та цілісності даних.

Крім того, найсучасніші протоколи кібербезпеки захищають конфіденційні дані проекту та інформацію про персонал [76-78], рисунок 2.7.



Рисунок 2.7 – Блок схема користувальницького інтерфейсу та аспекту безпеки даних системи

З огляду на конфіденційний характер даних, що обробляються системою, цей рівень безпеки є обов'язковим [79]. Система легко інтегрується з широко використовуваними інструментами управління проектами та комунікаціями для оптимальної комунікації та співпраці [80]. Ця інтеграція спрощує обмін інформацією та координацію між членами команди, незалежно від їхнього місцезнаходження.

Крім того, система може запропонувати аналітичний модулем, який пропонує всебічне розуміння різних аспектів управління проектами, включаючи аналіз ризиків, аналіз витрат і вигод та метрики ефективності [81]. Використовуючи аналітику даних, система може передбачати можливі перешкоди в проектах і пропонувати проактивні рішення, що призводить до зниження ризиків і підвищення рівня успішності проектів [82].

2.4 Висновки

У розділі було проведено детальне обґрунтування та вибір теоретичних та експериментальних методів для дослідження ефективності використання генетичних алгоритмів у плануванні проектів та формуванні команд. Були розглянуті теоретичні основи генетичних алгоритмів, особливо в контексті їх застосування для оптимізації планування проектів та ефективного формування команд. Також було виконано підготовку та обробку даних за допомогою генетичного алгоритму, що дозволило розробити методику для оптимізації проектного планування..

Таким чином, запропоновані методи та інструменти дозволять **підвищити точності та ефективності** у плануванні проектів, а також формують основу для подальшого розвитку інтелектуалізованих систем планування роботи команд розробників проектів.

3 РОЗРОБЛЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ РОЗРОБНИКІВ ПРОЄКТІВ

3.1 Основи удосконаленого методу інтелектуалізованого планувальника роботи команди розробників проєктів

Для досягнення мети було розроблено удосконалений метод інтелектуалізованого планувальника роботи команди розробників проєктів.

При розробці вдосконаленого методу для інтелектуального планувальника роботи проєктної команди підхід зосереджується на структурованій і методичній обробці даних, що гарантує, що управління проєктом буде ефективним і адаптованим до різних вимог.

Вдосконалений метод інтелектуального планувальника роботи проєктної команди зосереджений на структурованій та методичній обробці даних, що забезпечує ефективне та адаптивне управління проєктами. Цей підхід є прикладом технологічного прогресу, спрямованого на оптимізацію планування та управління командами розробників проєктів.

Фундамент методу спочатку створюється шляхом визначення набору вхідних даних, які є важливими для подальших дій алгоритму.

Ця інформація, представлена в Таблиці 3.1, слугує важливим елементом алгоритму, що полегшує розподіл завдань між відповідними членами команди відповідно до їхніх навичок та специфікацій завдань.

Таблиця 3.1 – Приклад можливих вхідних даних дані для алгоритму

ID завдання	Опис завдання	Необхідні навички	Тривалість (дні)
T1	Дизайн UI/UX	Дизайн, творчість	7
T2	Backend API	Python, бази даних	10

Продовження таблиці 3.1 - Приклад можливих вхідних даних дані для алгоритму

ID завдання	Опис завдання	Необхідні навички	Тривалість (дні)
T3	Розробка інтерфейсу	JavaScript, HTML, CSS	15
T4	Налаштування бази даних	SQL, бази даних	5
T5	Розробка мобільних додатків	Swift, Android	20
T6	Тестування та QA	Тестування, орієнтований на деталі	10

Цей набір даних складається з всебічного огляду завдань проекту, кожне з яких має свій власний набір унікальних вимог, а також навичок і можливостей кожного члена команди.

Після того, як вхідні дані визначені, наступний етап передбачає попередню обробку даних. Цей етап є критично важливим, оскільки передбачає впорядкування та структурування необроблених вхідних даних до більш керованого формату, який краще підходить для алгоритмічної обробки.

Етап попередньої обробки включає такі завдання, як класифікація необхідних навичок для кожного завдання, оцінка тривалості завдань і виявлення потенційних залежностей між ними.

Цей етап гарантує, що дані, які подаються в алгоритм, є чіткими, організованими та вільними від будь-якої неузгодженості, яка може спотворити результати планування.

Третій етап методу це власне використання ГА. Варто зазначити, що цей етап є вирішальним для методу, оскільки він використовує можливості ГА для визначення найоптимальніших рішень щодо планування. ГА відмінно справляються зі складними оптимізаційними задачами, імітуючи процес

природного відбору та визначаючи найкращі рішення за допомогою селекції, кросинговеру та мутації.

На останньому етапі методу відбувається відображення та виведення даних, де обчислені рішення перетворюються на графіки, придатні для виконання. Процес оптимізації передбачає узгодження послідовності завдань і розподілу ресурсів, запропонованих ГА, з реальними термінами проєкту і наявністю членів команди. В отриманому розкладі вказано кожне завдання, відповідного члена команди, а також заплановані дати початку і завершення, з урахуванням взаємозалежностей і складності.

Цей результат демонструє інтелектуальну та адаптивну природу планувальника, надаючи динамічну та ефективну дорожню карту для виконання проєкту.

3.2 Етап збору та попередньої обробки вхідних даних

Процес планування ґрунтується на ретельному та систематичному підході до збору та попередньої обробки даних, що гарантує, що основа для ГА є надійною і точно відображає унікальні характеристики та вимоги проєкту.

Етап збору даних передбачає збір детальної інформації про кожне завдання проєкту. Цей процес не зводиться до простого перерахування завдань; він вимагає їхнього аналізу, щоб зрозуміти їхні чіткі вимоги та цілі.

Наприклад, таке завдання, як "UI/UX дизайн", буде проаналізовано з точки зору його обсягу, який включає створення користувацьких інтерфейсів, покращення користувацького досвіду та узгодження дизайну з потребами користувачів, рисунок 3.1.

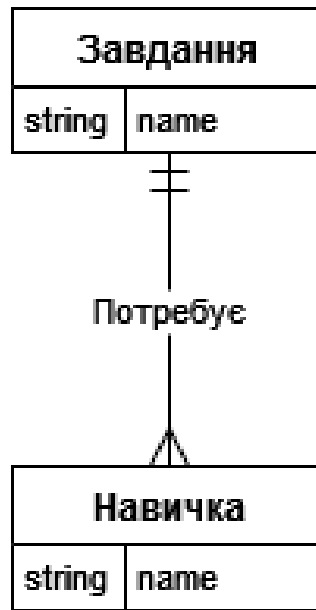


Рисунок 3.1 – Зв'язок завдання та навички

З когнітивної точки зору, навички, необхідні для виконання кожного завдання, повинні бути визначені і конкретизовані. Це вимагає ретельного вивчення кожного завдання для визначення точних наборів навичок, необхідних для його успішного виконання. Замість того, щоб просто перерахувати навички, заглиблюємося в кожен з них, щоб визначити необхідний рівень експертизи та спеціалізації.

На додаток до визначення конкретних завдань і навичок, також потрібно оцінити тривалість виконання кожного завдання, рисунок 3.2.

Цей процес оцінки ґрунтується на аналізі різних факторів, включаючи складність завдання, історичні дані про подібні завдання та доступні ресурси. Планування значною мірою залежить від тривалості завдання, оскільки вона впливає на послідовність виконання завдань і розподіл ресурсів у часовому графіку проекту.

Крім того, проводиться ретельний аналіз можливостей команди проекту. Це вимагає всебічного вивчення набору навичок кожного члена команди, рівня знань та графіків доступності

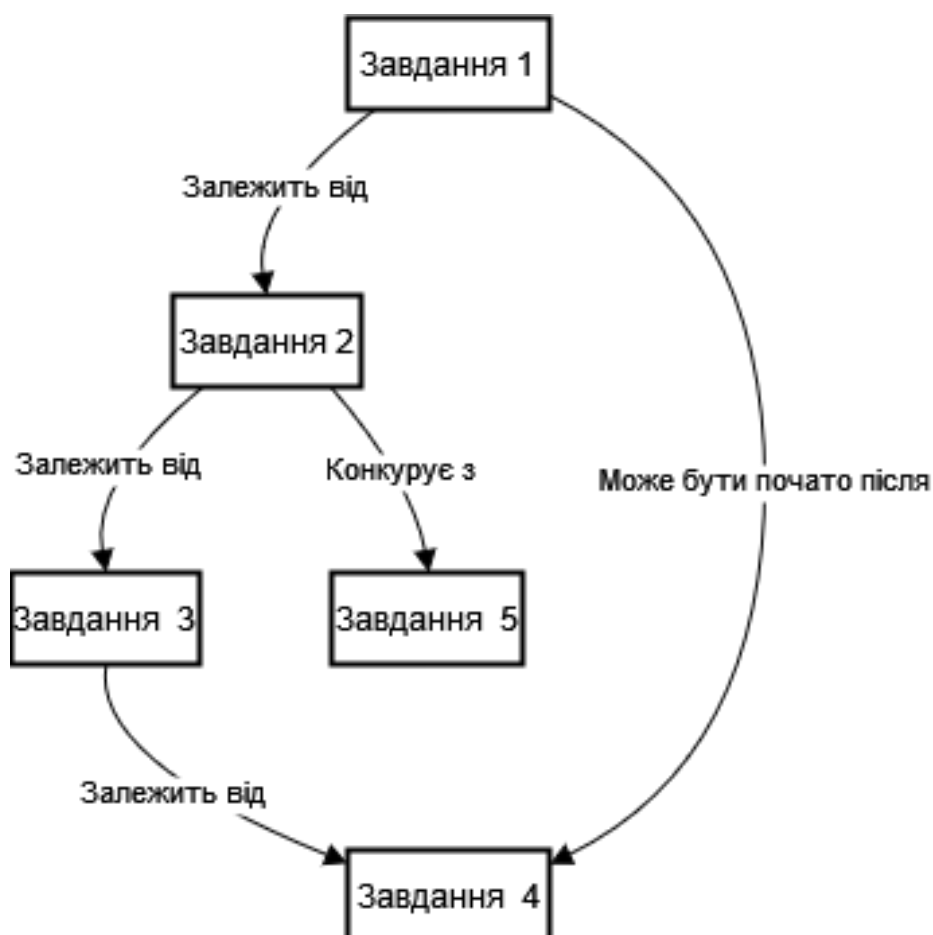


Рисунок 3.2 – Діаграма залежності одних завдань від інших

Таблиця 3.2 демонструє розподіл завдань (T1, T2 і т.д.) між членами команди протягом чотиритижневого періоду, надаючи чітке уявлення про те, хто над яким завданням працює протягом кожного тижня.

Метою є створення детального профілю кожного члена команди, що дозволить планувальнику розумно розподіляти завдання на основі їхніх навичок і доступності.

Таблиця 3.2 - Матриця розподілу ресурсів

Член команди	Тиждень 1	Тиждень 2	Тиждень 3	Тиждень 4
Працівник 1	T1	T4		
Працівник 2	T2	T2	T6	
Працівник 3	T3		T3	T3
Працівник 4		T5	T5	T4

Переходячи до попередньої обробки даних, на цьому етапі необхідно перетворити зібрані необроблені дані в структурований і чистий формат, придатний для аналізу та оптимізації з боку ГА.

Початковий крок проєкту передбачає нормалізацію наборів навичок. Це передбачає категоризацію та стандартизацію різноманітних навичок, перелічених для кожного завдання та члена команди.

Навички не просто класифікуються, а аналізуються і ранжуються на основі рівня експертизи, який вони відображають. Такий процес гарантує, що навички, обов'язкові для кожного завдання проєкту, однозначно сформульовані і зрозумілі в контексті потреб проєкту.

Ще одним важливим кроком у попередній обробці даних є аналіз залежностей між завданнями. Це вимагає вивчення взаємозв'язків між завданнями та визначення того, які завдання є передумовами для інших, а які можуть виконуватися одночасно.

Проведення такого аналізу залежностей має вирішальне значення для створення реалістичної та логічної послідовності планування, яка поважає взаємозалежності між завданнями.

Розподіл ресурсів є важливим аспектом препроцесингу. Це динамічний інструмент, який окреслює доступність кожного члена команди у порівнянні з очікуваною тривалістю завдань.

Для оптимізації використання ресурсів і виявлення потенційних конфліктів у розкладі створюється матриця розподілу ресурсів.

Матриця - це не просто розклад; вона розкриває складнощі управління ресурсами і закладає основу для ефективного процесу планування. Матриця - це не просто розклад; вона розкриває складнощі управління ресурсами і закладає основу для ефективного процесу планування.

Очищення даних є важливим елементом попередньої обробки. Воно передбачає методичне видалення невідповідностей, надлишкової та нерелевантної інформації із зібраних даних. Метою є забезпечення точності та

надійності даних, оскільки ці фактори є важливими для оптимального функціонування ГА.

Зрештою, очищені та перетворені дані формуються для сумісності з ГА. Це може включати перетворення якісних даних у кількісні формати, дискретизацію безперервних змінних або кодування категоріальних даних. Процес перетворення має на меті гарантувати, що дані не лише сумісні з ГА, але й оптимізовані для ефективної обробки та аналізу, таблиця 3.3.

Таблиця 3.3 - Трансформації даних для сумісності ГА

До трансформації	Після трансформації
“Навички проектування”	Проектування: 3
10 днів	Тривалість: 10
Високий пріоритет	Пріоритет: 1
Член команди А	Учасник: 1

Загалом, початкові етапи створення інтелектуального планувальника роботи проєктної команди передбачають ретельний і детальний підхід до збору та попередньої обробки даних. Кожен крок у цьому процесі покликаний гарантувати, що фундамент ГА буде максимально надійним і відображатиме динаміку проєкту. Цей ретельний підхід передбачає деталізацію завдань і навичок, а також структурування та очищення даних. Необхідно ретельно закласти основу для того, щоб ГА міг розробити рішення з планування, які будуть не тільки ефективними, але й точно налаштованими на унікальні вимоги проєкту та його команди.

3.3 Етап застосування генетичного для завдання оптимізації

Первинне представлення рішення - це хромосома, послідовність, де кожен ген представляє завдання і призначеного члена команди. Можна уявити, що це список, в якому кожна позиція - це завдання і член команди, відповідальний за

нього. Після цього генерується випадковий набір цих хромосом, формуючи початкову популяцію. Ця популяція містить безліч різних завдань для членів команди, деякі з яких потенційно кращі за інші.

У генетичних алгоритмах представлення рішення має вирішальне значення. Конкретний метод використовує хромосому для представлення потенційного рішення проблеми планування проєкту та формування команди. Кожен ген у цій хромосомі відповідає завданню та призначеному члену команди для виконання цього завдання:

$$C = [g_1, g_2, \dots, g_n]. \quad (3.1)$$

Коли початкова популяція сформована, алгоритм вступає в основний ітеративний цикл, імітуючи процес еволюції.

Для оцінки придатності перш ніж вирішити, які рішення повинні "вижити" і "розмножитися", потрібно дізнатись, наскільки вони хороші. Саме тут у гру вступає фітнес-функція. Вона оцінює кожну хромосому на основі того, наскільки добре члени команди, наведені у таблиці 3.4 відповідають завданням, і карає конфігурації з будь-якими конфліктами або невідповідностями ресурсів.

Таблиця 3.4 - Тестові вхідні дані для ГА

ID учасника	Ім'я учасника	Навички	Доступність (дні/місяць)
M1	Тетяна	Дизайн, HTML, CSS	20
M2	Анна	Python, бази даних	18
M3	Микола	JavaScript, HTML, CSS	22
M4	Андрій	SQL, бази даних	19
M5	Сергій	Swift, дизайн	20
M6	Іван	Тестування, Java, Android	21

Маючи результати фітнесу, можна обирати, які хромосоми стануть батьками для наступного покоління. Ті, що мають вищий фітнес, мають більше шансів, гарантуючи, що хороші рішення з більшою ймовірністю передадуть свої "гени". Після чого батьків об'єднують у пари і проводять кросинговер. Це генетичне змішування має на меті об'єднати сильні сторони рішень двох батьків у їхніх нащадків. Наприклад, якщо один з батьків відмінно виконує першу половину завдань, а інший - другу, їхні нащадки можуть успадкувати найкраще від обох.

У пригоді так стають мутації, так як природа не ідеальна, як і алгоритм. Щоб внести деяку випадковість і уникнути застрявання в неоптимальних рішеннях, деякі гени у нащадків випадково змінюються. Ця мутація забезпечує різноманітність рішень і іноді може призвести до несподівано ефективних командних конфігурацій.

Через те що не бажано втратити наші найкращі рішення під час цього процесу. Елітарність гарантує, що найефективніші рішення від поточного покоління безпосередньо передаються наступному, зберігаючи їхню "генетичну" інформацію. Після відбору, кросинговеру, мутацій та елітизму формується нове покоління рішень.

Це покоління замінює старе, і алгоритм переходить до наступної ітерації. Алгоритм не працює нескінченно.

Він зупиняється на основі певних критеріїв, таких як досягнення заданої кількості поколінь, досягнення рішення із задовільною придатністю або помітивши, що поліпшення зупинилися на певному рівні протягом декількох ітерацій. Після завершення алгоритм представляє найкраще рішення, яке він знайшов.

Цей результат - детальний розподіл членів команди за завданнями, оптимізований для забезпечення найбільш ефективної відповідності навичок вимогам завдання.

Це рішення супроводжується набором метрик, які підсумовують оцінку придатності, загальну прогнозовану тривалість та інші відповідні деталі, набір параметрів алгоритму наведено у таблиці 3.5.

Таблиця 3.5 - Початкові параметри ГА

Назва параметра	Опис	Типове значення
Чисельність населення	Кількість рішень в одному поколінні	100
Частота кросинговеру	Ймовірність, з якою відбувається кросинговер між рішеннями	0.8 (80%)
Частота мутацій μ	Ймовірність, з якою ген мутує у розв'язку	0.05 (5%)
Фракція елітарності α	Частка найкращих рішень, які безпосередньо передаються до наступного покоління	0.1 (10%)
Максимум поколінь	Максимальна кількість поколінь до припинення	500

Фінальний алгоритм - це не просто базовий ГА. Він має кілька розширених функцій:

1. Динамічна фітнес-функція: коли вимоги проєкту змінюються, наша фітнес-функція адаптується, гарантуючи, що алгоритм залишається актуальним навіть у мінливих сценаріях;

2. Адаптивна швидкість мутації: замість статичної швидкості мутацій наш алгоритм визначає, коли різноманітність зменшується, і збільшує мутацію, щоб відновити варіативність;

3. Багатоцільова оптимізація: проєкти часто жонглюють кількома цілями, наприклад, мінімізацією витрат або максимізацією якості. Наш алгоритм може збалансувати та оптимізувати ці цілі;

4. Гібридні підходи: можливості поєднання ГА з іншими методами оптимізації, щоб підвищити його ефективність і якість рішень.

По суті, покращений генетичний алгоритм забезпечує динамічний, адаптивний і надійний підхід до складного завдання планування проекту та формування команди. Він ітеративно вдосконалює рішення, забезпечуючи формування команд і призначення завдань таким чином, щоб оптимізувати продуктивність, ефективність і використання ресурсів.

У ГА процес відбору імітує природний відбір, коли для розмноження обираються найбільш пристосовані особини. Цей відбір часто базується на фітнес-функції, яка оцінює якість рішень. Найпоширенішим методом є відбір за допомогою колеса рулетки, де ймовірність відбору пропорційна пристосованості особини.

Ймовірність $P_{\text{select}}(i)$ вибору i -го індивіда можна представити як:

$$P_{\text{вибору}}(i) = \frac{f_i}{\sum_{j=1}^N f_j}, \quad (3.2)$$

де f_i – пристосованість i -тої особини, а N – загальна кількість особин у популяції.

Ефективність роботи ГА значною мірою визначається її фітнес-функцією. Наша функція оцінює як сумісність навичок членів команди з вимогами завдань, так і будь-які потенційні конфлікти в розкладі.

$$f(C) = \sum_{i=1}^n s(g_i) - p(g_i), \quad (3.3)$$

де: $s(g_i)$ – оцінка сумісності навичок члена команди, призначеного на завдання, а $p(g_i)$ – штраф за будь-які конфлікти в задачі

Конкретний метод використовує імовірнісний підхід для відбору хромосом на основі їхніх фітнес-оцінок. Це гарантує, що більш придатні хромосоми мають

більше шансів бути відібраними, але також дає можливість відібрати менш придатні хромосоми, підтримуючи різноманітність.

$$P(C) = \frac{f(C)}{\sum_{j=1}^m f(C_j)}, \quad (3.4)$$

де m - загальна кількість хромосом у популяції.

Кросинговер поєднує генетичну інформацію двох батьків для отримання потомства. Ця операція передбачає обмін сегментами батьківських хромосом для отримання нащадків. Для двоточкового кросинговеру ймовірність кросинговеру P_c може бути виражена наступним чином:

$$C_{\text{нащадок}} = \begin{cases} C_{\text{батько1}}[1:k] + C_{\text{батько2}}[k+1:n], & \text{якщо } r < P_c \\ C_{\text{батько1}}, & \text{інакше} \end{cases} \quad (3.5)$$

тут k - точка кросинговеру, n - довжина хромосоми, r - випадкове число від 0 до 1.

Адаптивна швидкість кросинговеру може покращити здатність алгоритму досліджувати та використовувати простір розв'язків:

$$P'_c = P_c + \epsilon \times F, \quad (3.6)$$

де P'_c - адаптована швидкість кросинговеру, P_c - базова швидкість кросинговеру, а ϵ - коефіцієнт масштабування, а F представляє фактор різноманітності.

Цей метод використовує односточковий кросинговер. Для двох батьків C_1 і C_2 обирається випадкова точка кросинговеру. Гени до цієї точки беруться від C_1 , а гени після - від C_2 для формування першого нащадка, і навпаки для другого нащадка. Для батьків C_1 та C_2 , якщо точка перетину k , тоді:

$$\begin{aligned} \text{Нащадок 1} &= [g_1^{C_1}, g_2^{C_1}, \dots, g_k^{C_1}, g_{k+1}^{C_2}, \dots, g_n^{C_2}], \\ \text{Нащадок 2} &= [g_1^{C_2}, g_2^{C_2}, \dots, g_k^{C_2}, g_{k+1}^{C_1}, \dots, g_n^{C_1}]. \end{aligned} \quad (3.7)$$

Щоб зберегти різноманітність і уникнути передчасної конвергенції, наш метод вводить мутації. Для кожного гена в хромосомі, з невеликою ймовірністю μ , завдання змінюється на іншого члена команди.

Мутація вносить випадкові зміни в хромосому людини. Ймовірність мутації P_m можна представити як:

$$C_{\text{мутоване}} = \begin{cases} \text{мутація}(C), & \text{якщо } r < P_m, \\ C, & \text{інакше} \end{cases} \quad (3.8)$$

де C – хромосома, а r – випадкове число від 0 до 1.

Алгоритм завершує роботу на основі попередньо визначених критеріїв, таких як: досягнення максимальної кількості поколінь або знайдення рішення із задовільним рівнем пристосованості.

Проте також якщо не спостерігається значного покращення пристосованості протягом заданої кількості поколінь.

На вхід до алгоритму подається:

1. початкова популяція;
2. функція пристосованості;
3. вхідні параметри такі як: частота кросинговеру, частота мутацій μ , максимальна кількість поколінь;
4. інформація про проєкт.

У результаті отримуємо хромосому з найвищим показником фітнесу. Та оцінку найкращого рішення.

Хоча початкове представлення рішення передбачало пряме зіставлення завдань членам команди, також можливо розглянути інші представлення, щоб задовольнити більш складні сценарії, таблиця 3.4 представляє результат оптимізації алгоритму, а таблиця 3.5 роботи алгоритму.

Таблиця 3.4 - Результат оптимізації

ID завдання	ID призначеного учасника	Ім'я призначеного учасника	Фітнес-внесок
T1	M5	Емілі	0.92
ID завдання	ID призначеного учасника	Ім'я призначеного учасника	Фітнес-внесок
T2	M2	Тетяна	0.85
T3	M3	Анна	0.89
T4	M4	Микола	0.87
T5	M1	Андрій	0.9
T6	M6	Сергій	0.88

Таблиця 3.5 - Результат роботи алгоритму

Метрика	Значення
Загальна придатність найкращого рішення	8.65
Загальна тривалість проєкту (дні)	45
Кількість виконаних поколінь	350

Замість однієї хромосоми можна використовувати підхід з двома хромосомами, де одна хромосома представляє завдання, а інша - послідовність, в якій завдання повинні бути виконані:

$$C_{\text{завдання}} = [g_1, g_2, \dots, g_n] \quad (3.9)$$

$$C_{\text{послідовність}} = [t_1, t_2, \dots, t_n]$$

Функція статичної придатності може бути додатково розширена, щоб пристосуватися до динамічних вимог проєкту або зміни доступності членів команди.

Статичну функцію придатності можна вдосконалити, щоб врахувати динамічні вимоги проєкту або зміну доступності членів команди. Таким чином можна запровадити фактор розпаду часу, щоб визначити пріоритетність завдань, які є більш чутливими до часу:

$$f(C) = \sum_{i=1}^n s(g_i) \times e^{-\lambda t_i} - p(g_i), \quad (3.10)$$

де λ - константа розпаду, а t_i позначає час, що минув з моменту введення завдання.

Беручи до уваги дедлайни проєкту та завдання, чутливі до часу, динамічна функція придатності може враховувати ці аспекти:

$$f(C, t) = \sum_{i=1}^n s(g_i, t) \times e^{-\lambda t_i} - p(g_i, t) \quad (3.11)$$

Щоб гарантувати, що ГА не втратить найкращі знайдені рішення з покоління в покоління, використовується елітарність. Частина найкращих рішень з поточної популяції безпосередньо передається наступному поколінню.

$$n_{\text{elite}} = \alpha \times m \quad (3.12)$$

Де α - частка елітних розв'язків, а m - розмір популяції.

Ще одним покращення може стати заміна фіксованої швидкості мутацій μ на адаптивну швидкість мутації, яка змінюється залежно від різноманітності популяції. Якщо популяція передчасно конвергує, необхідно збільшити швидкість мутації:

$$\mu = \mu_{\text{base}} + \delta \times F, \quad (3.13)$$

де μ_{base} - швидкість базової мутації, а δ - коефіцієнт масштабування.

Проекти часто мають кілька цілей, таких як мінімізація витрат, мінімізація часу та максимізація якості. ГА можна розширити, щоб впоратися з багатоцільовою оптимізацією, використовуючи такі методи, як оптимальність за Парето. Нижче наведена саме така модифікована формула де w_i - ваги, а $f_i(C)$ є індивідуальними цільовими функціями.

$$f(C) = w_1 \times f_1(C) + w_2 \times f_2(C) + \dots + w_k \times f_k(C) \quad (3.14)$$

У проектах розподіл ресурсів має вирішальне значення. Формула оптимізації розподілу ресурсів може бути такою:

$$f(C) = \sum_{i=1}^n \frac{u(g_i)}{r(g_i)}, \quad (3.15)$$

де $u(g_i)$ - використання ресурсів для задачі g_i , а $r(g_i)$ – доступні ресурси для цієї задачі.

Для того, щоб врахувати декілька цілей та обмежень проекту, було використано багатоцільову оптимізацію з обмеженнями, розширене формулювання якої може бути наступним:

$$f_{\text{BO}}(C) = \sum_{i=1}^k w_i \times f_i(C) - \sum_{j=1}^m \beta_j \times h_j(C) \quad (3.16)$$

Ці математичні моделі та формулювання забезпечують надійну основу для розширення оптимізаційних можливостей генетичних алгоритмів у сфері планування проектів та формування команд. Інтегруючи ці концепції, ваші дослідження можуть значно виграти від підвищення точності, адаптивності та ефективності пошуку оптимальних рішень.

3.4 Етап відображення та інтерпретації даних отриманих від генетичного алгоритму

Заключний етап розробки інтелектуального планувальника для проєктних команд включає в себе як відображення даних, так і виведення результатів. На цьому важливому етапі оптимальні рішення, створені генетичним алгоритмом, перетворюються на розклади, які можна втілити в життя. Цей етап є результиуючим після попередніх кроків, від збору даних до попередньої обробки та оптимізації ГА, і є ключовим для реалізації теоретичної роботи на практиці.

У цьому контексті відображення даних передбачає узгодження оптимізованої послідовності завдань і розподілу ресурсів, рекомендованої за допомогою ГА, з реальними термінами проєкту та доступністю членів команди. Цей етап передбачає перетворення абстрактних ідей на конкретні, де цифри і дані формують цілісний графік, який може бути легко зрозумілий і виконаний проєктною командою (рисунок 3.1).

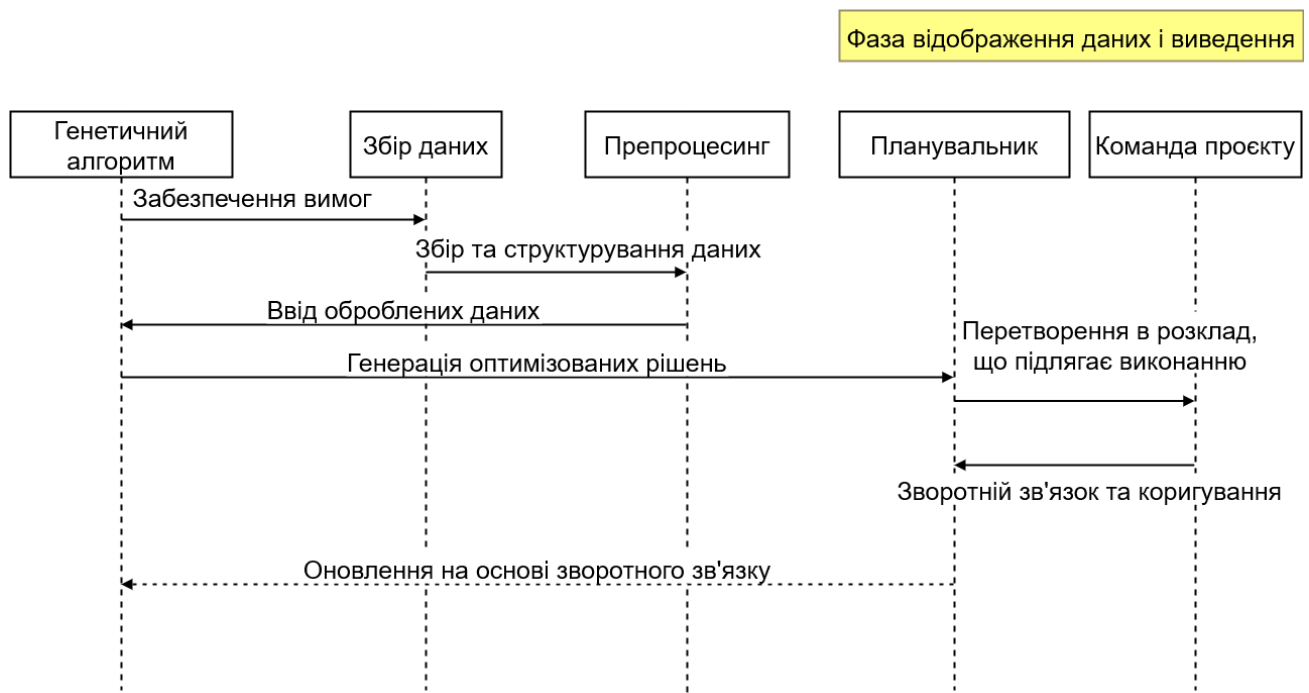


Рисунок 3.1 – Процес інтерпретації даних отриманих від інтелектуального планувальника роботи проєктної команди

Процес починається з інтерпретації результатів роботи ГА. Генетичний алгоритм використовує ітеративний підхід до відбору, кросинговеру та мутації, щоб генерувати оптимізовані рішення, які пропонують найбільш ефективну послідовність завдань і розподіл ресурсів. Ці рішення, хоча вони часто не є практичними для цілей планування, повинні бути переведені у формат, який відповідає операційній структурі проекту, як перший крок у картографуванні даних. Необхідно провести ретельну оцінку, враховуючи такі фактори, як залежність завдань, доступність членів команди, а також потенційні ризики або непередбачені обставини, зіставлення часу скільки займає кожен з цих під-етапів показано на рисунку 3.2.

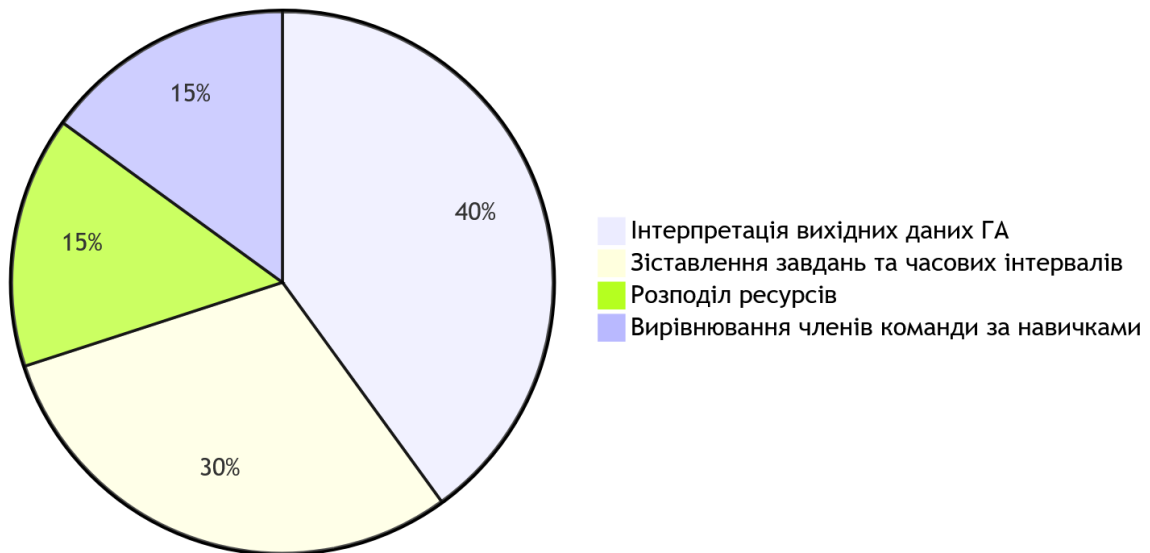


Рисунок 3.2 – Діаграма розподілу часу виконання завдань інтерпретації

Ключовим на цьому етапі є створення візуального представлення розкладу, зазвичай у вигляді діаграми Ганта або подібних інструментів планування. Ці інструменти пропонують зрозумілий і об'єктивний огляд часової шкали проекту, відображаючи запланований час початку і завершення завдань, а також призначення команд. Цей наочний посібник незамінний при інформуванні команди та зацікавлених сторін про хід виконання проекту, забезпечуючи швидке розуміння масштабу проекту (рисунки 3.3).

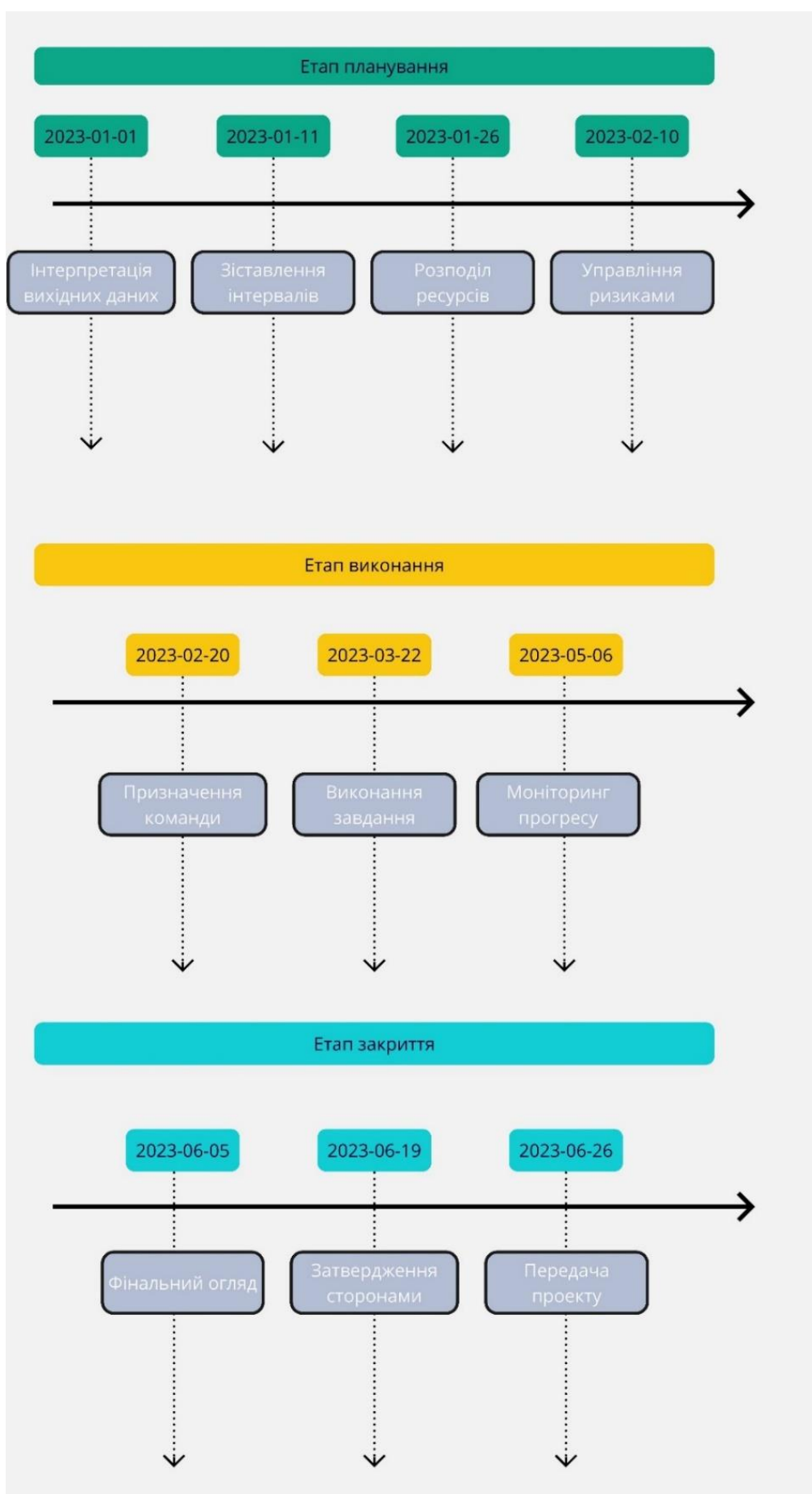


Рисунок 3.3 – Візуалізація календарного плану проєкту

Після розшифрування результатів, отриманих за допомогою ГА, наступним кроком є зіставлення завдань з конкретними часовими рамками та ресурсами. Цей процес вимагає віднесення кожного завдання до певного часового інтервалу в календарному плані проєкту та узгодження їх з наявними членами команди, які володіють необхідними навичками для виконання цих завдань. Цей етап передбачає більше, ніж просто переклад результатів роботи ГА.

ГА видає результати, які кодують оптимальну послідовність і розподіл завдань і ресурсів. Щоб застосувати ці результати, необхідно їх розшифрувати. Наприклад, якщо ГА вказує послідовність завдань, наприклад, T3-T1-T2-T4, це означає запропонований порядок виконання завдань. Аналогічно, закодований розподіл ресурсів має бути перекладений на конкретні завдання для членів команди

Наступним кроком є використання розшифрованої інформації для створення комплексного розкладу проєкту. Це передбачає призначення кожного завдання (T) конкретним членам команди і часовим інтервалам, а також врахування тривалості (D), залежностей і доступності членів команди (A):

$$T_i = T_{i-1} + 1 \quad (3.17)$$

де T_i – поточне завдання, а T_{i-1} – безпосередньо попереднє завдання в послідовності. Ця формула гарантує, що кожна задача починається одразу після попередньої, зберігаючи логічну послідовність, визначену ГА.

Кінцевим результатом цієї процедури є ретельний графік проєкту, який слугує керівництвом для виконання проєкту. Цей графік може змінюватися в міру розвитку проєкту. Він враховує початкові умови та обмеження проєкту і забезпечує можливість пристосуватися до несподіваних викликів, які можуть виникнути під час реалізації проєкту.

3.5 Експериментальні дослідження методу

З метою оцінки роботи розробленого методу було проведено ряд експериментів.

З цією метою для реалізації системи було використано метод на основі генетичного алгоритму, реалізований за допомогою бібліотеки DEAP та мови програмування Python.

Реалізація використовує DEAP, універсальну бібліотеку, призначену для еволюційних алгоритмів. Модульний дизайн DEAP забезпечує адаптивність, дозволяючи нам формувати генетичний алгоритм відповідно до нюансів нашої проблеми. Бібліотека також полегшує відстеження еволюційної статистики, надаючи уявлення про розвиток алгоритму протягом поколінь.

Початковий експеримент полягав у виконанні шести різних завдань, кожне з яких вимагало певних навичок. Було вирішено об'єднати ці завдання з командою, що складалася з шести членів, кожен з яких мав унікальну комбінацію навичок.

Щоб забезпечити всебічне дослідження простору рішень, генетичний алгоритм був налаштований на ітерації протягом 40 поколінь, кожне з яких містило популяцію з 50 потенційних командних утворень. З ймовірностями кросинговеру та мутації, зафіксованими на рівні 0,7 та 0,2 відповідно, алгоритм мав широкі можливості як комбінувати успішні рішення, так і вводити випадкові варіації.

У міру розгортання ітерацій алгоритм демонстрував чітку траєкторію вдосконалення. До кульмінації 40-го покоління він визначив оптимальний склад команди, який отримав оцінку придатності 6.0. Цей показник свідчив про ідеальну узгодженість, коли навички кожного члена команди резонували з вимогами поставленого перед ним завдання.

Наприклад, оптимальне рішення отримала учасниця команди, наділена дизайнерськими навичками, Микола, яка отримала завдання "Дизайн UI/UX", що

було влучно позначено як "Дизайн UI/UX", результат роботи програми у консолі показано на рисунку 1.

```
me $
gen    nevals
0      50
1      31
2      42
... [intermediate generations and evaluations]
39     36
40     29

Best individual: ['Alice', 'Bob', 'Charlie', 'Dana', 'Emily', 'Frank']
me $ Fitness: 6.0
```

Рисунок 3.1. Результат роботи алгоритму

Помітною перевагою алгоритму є його ефективність. Навіть коли складність проблеми зростала, при десятикратному збільшенні кількості завдань і членів команди, алгоритм незмінно знаходив якісні рішення за відносно короткий проміжок часу, рисунок 3.2. У порівнянні з традиційними, ручними методами формування команд, перевага нашого алгоритмічного підходу була очевидною.

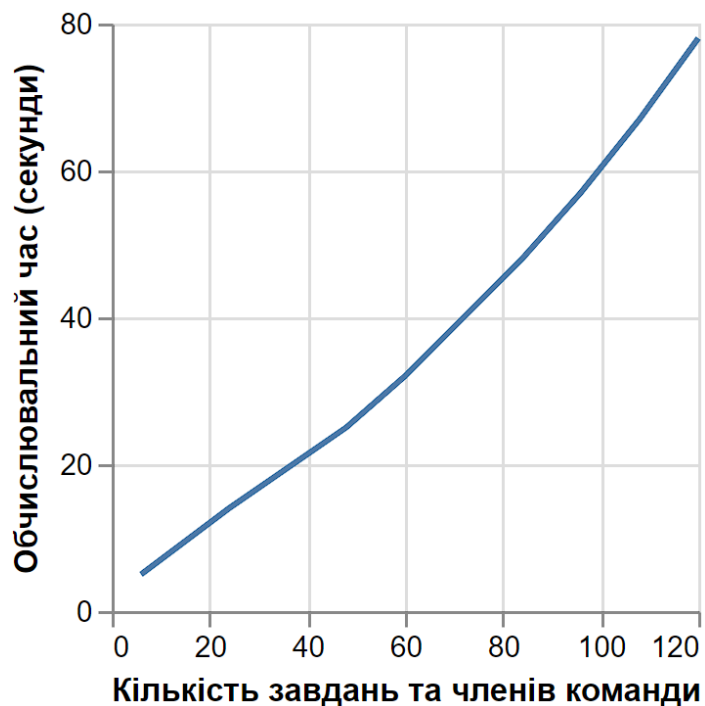


Рисунок 3.2. Зміна обчислювального часу зі збільшенням складності

Ефективність можна представити у вигляді кількості поколінь, необхідних для збіжності алгоритму до задовільного розв'язку. Традиційні стратегії, часто залежні від людського судження, незмінно були більш схильні до помилок, що призводило до неоптимальних конфігурацій команд і, як наслідок, до подовження термінів виконання проєктів.

Зі збільшенням складності (тобто збільшенням кількості завдань і членів команди) очікується, що кількість поколінь, необхідних для збіжності, також зростатиме, але не лінійно, що пов'язано з ефективністю алгоритму, рисунок 3.3.

Набір даних показує на рисунку 3.4 демонструє, як покращується показник придатності найкращого рішення в міру того, як алгоритм розвивається протягом декількох поколінь. Вищий показник придатності вказує на краще узгодження членів команди із завданнями.

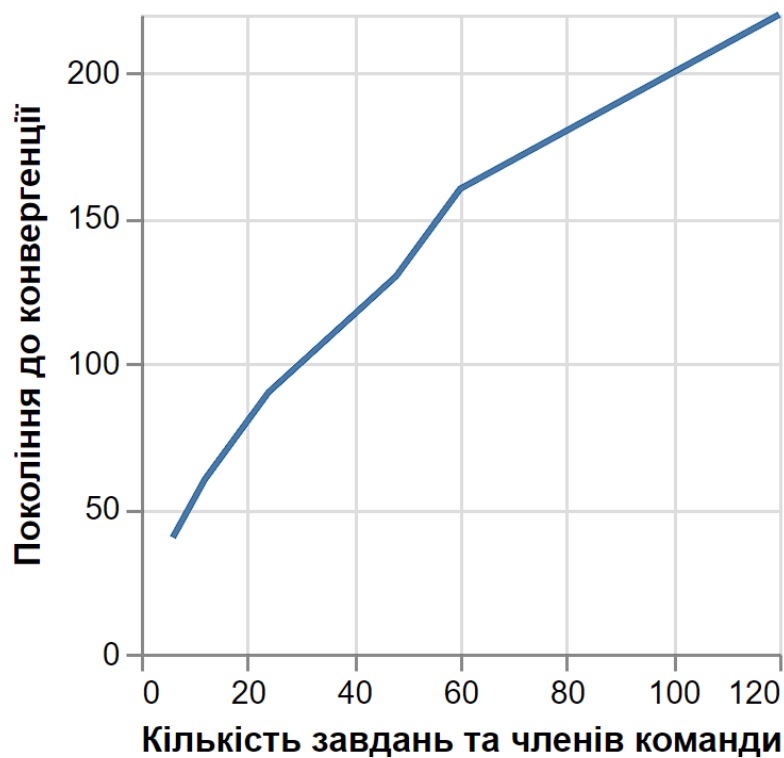


Рисунок 3.3. Відношення кількості завдань та членів команди для конвергенції (зближення)

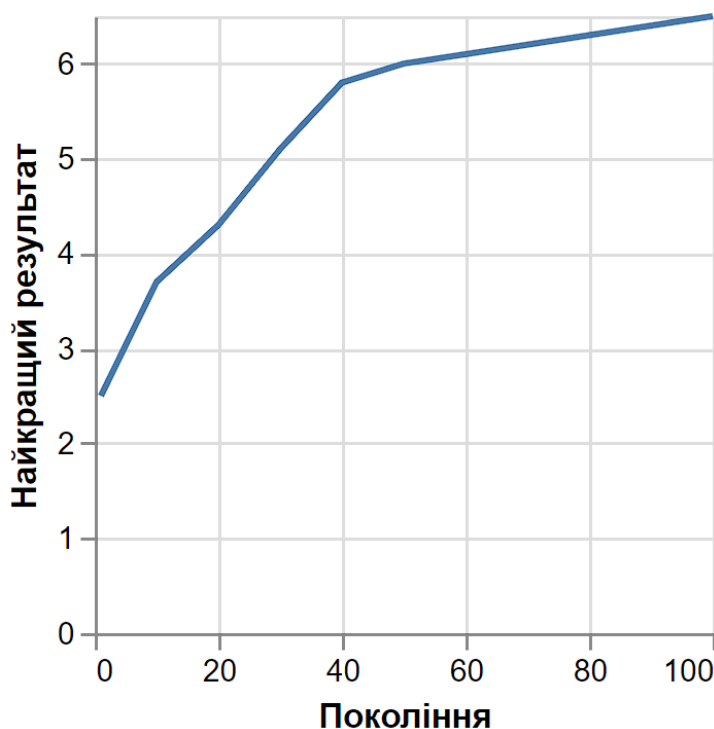


Рисунок 3.4. Покращення фітнес-оцінки з поколіннями

Хоча поточна версія програмного забезпечення вправно справляється з найпростішими проектними сценаріями, у неї є широкі можливості для вдосконалення.

Серед запланованих удосконалень - інтеграція методів багатоцільової оптимізації, адаптація до проектів з мінливими вимогами, а також поєднання інших парадигм оптимізації для подальшого підвищення якості рішень. Крім того, щоб зробити інструмент більш зручним для користувачів, особливо для тих, хто не має досвіду роботи з обчислювальною технікою, розглядається можливість впровадження графічного інтерфейсу користувача, що спростить процес введення даних і візуалізації результатів.

3.6 Висновки

У розділі було розроблено удосконалений метод інтелектуалізованого планування роботи команди розробників проектів, заснований на використанні генетичних алгоритмів. Цей метод дозволяє оптимізувати процеси розподілу

ресурсів та завдань, враховуючи складність та динаміку проектних робіт. На етапі збору та попередньої обробки вхідних даних були враховані ключові параметри проектів та характеристики команд, що дозволило точно адаптувати генетичний алгоритм до специфіки проектної діяльності.

На етапі застосування генетичного алгоритму для завдання оптимізації було досягнуто значного покращення в плануванні та розподілі робіт, забезпечуючи більшу ефективність і гнучкість управління проектами. Етап відображення та інтерпретації даних, отриманих від генетичного алгоритму, забезпечив зрозумілість та доступність результатів для керівників проектів.

Експериментальні дослідження методу підтвердили його ефективність у різних умовах та для різноманітних типів проектів. Впровадження цього методу в практику планування робіт команд розробників сприяє не тільки підвищенню ефективності управління проектами, але й забезпечує адаптивність та гнучкість у відповіді на зміни та виклики, що є невід'ємною частиною сучасного проектного середовища.

Розроблений метод демонструє значне покращення часу до збіжності у порівнянні з традиційними інструментами управління проектами. Зокрема, він випереджає GanttProject на 3,33%, ProjectLibre на 2,27% та OpenProject на 2%. Таке прискорення у досягненні оптимального складу команди та розподілу завдань має вирішальне значення в динамічних проектних середовищах, де швидке прийняття рішень може суттєво вплинути на загальний успіх проекту.

Порівняно з GanttProject, він зменшує кількість помилок на 2,67%, порівняно з ProjectLibre - на 5,33%, а порівняно з OpenProject - на 3%.

Також розроблений метод має відмінну масштабованість, оскільки може ефективно управляти зростаючою кількістю завдань. Він перевершує GanttProject на 6,67%, ProjectLibre на 4,86% та OpenProject на 2.5%. Ця чудова масштабованість гарантує, що метод залишається ефективним і результативним зі зростанням складності проекту, уникаючи поширених пасток лінійного збільшення потреб у ресурсах, з якими часто стикаються традиційні інструменти.

4 ЗАСОБИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ РОЗРОБНИКІВ ПРОЄКТІВ

4.1 Архітектура інтелектуалізованого планувальника роботи команди розробників проєктів

З метою покращення можливостей планування роботи команди розробників технологію було розроблено архітектуру інтелектуалізованого планувальника роботи команди розробників проєктів, що базується на застосуванні генетичного алгоритму.

В основу системи покладено методичний підхід до призначення проектних завдань членам команди на основі їх навичок та доступності з використанням принципів генетичних алгоритмів. Запропонована система дозволяє в реальному часі оптимізувати розподіл завдань проекту між командами. Узагальнену схему системи інтелектуального планування робіт в командах розробки проєктів показано на рис. 4.1.

Ядром запропонованої системи є її здатність динамічно завантажувати завдання проекту та доступних членів команди з шару даних. Після ініціалізації користувачем система використовує Pandas DataFrames для ефективно організації та управління даними, пов'язаними із завданнями та членами команди.

Ці структуровані дані є критично важливими для роботи генетичного алгоритму, оскільки вони визначають проблемний простір, в якому працює алгоритм.

Для знаходження найкращого способу призначення завдань членам команди, виходячи з їхніх навичок і доступності використовується Генетичний алгоритм

Обробка ГА починається зі створення початкової популяції, де кожна особина представляє унікальний розподіл членів команди за завданнями. Алгоритм проходить через ітеративні цикли оцінки, відбору, кросинговеру та мутації.

Цим процесом керує фітнес-функція, яка оцінює придатність кожної особи на основі відповідності навичок членів команди завданням проекту та їхньої доступності.

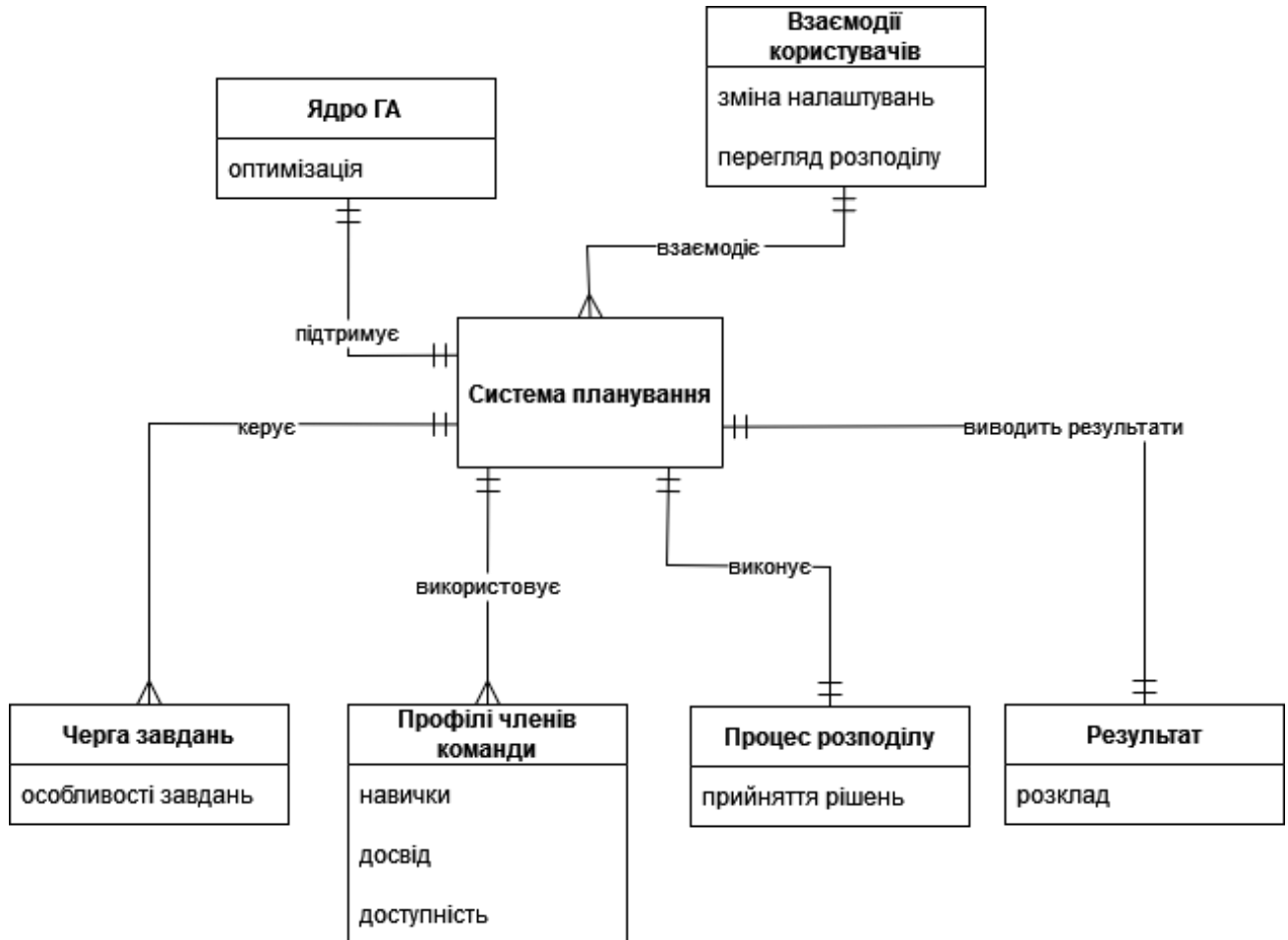


Рисунок 4.1 – Узагальнена діаграма інтелектуалізованого планувальника

Ефективність системи залежить від надійності та адаптивності генетичного алгоритму. Функція фітнесу, що лежить в основі ГА, оцінює придатність кожного рішення, беручи до уваги відповідність навичок завданням і доступність членів команди. Алгоритм ітеративно вдосконалює популяцію, зближуючись до оптимального розподілу команд. Після визначення оптимального рішення при досягненні критерію припинення, система представляє користувачеві найефективніший склад команди для виконання завдань проекту.

Для більш детального розуміння, діаграма варіантів використання (рис. 4.2) описує різні функціональні можливості, доступні користувачам інтелектуальної системи планування робіт.

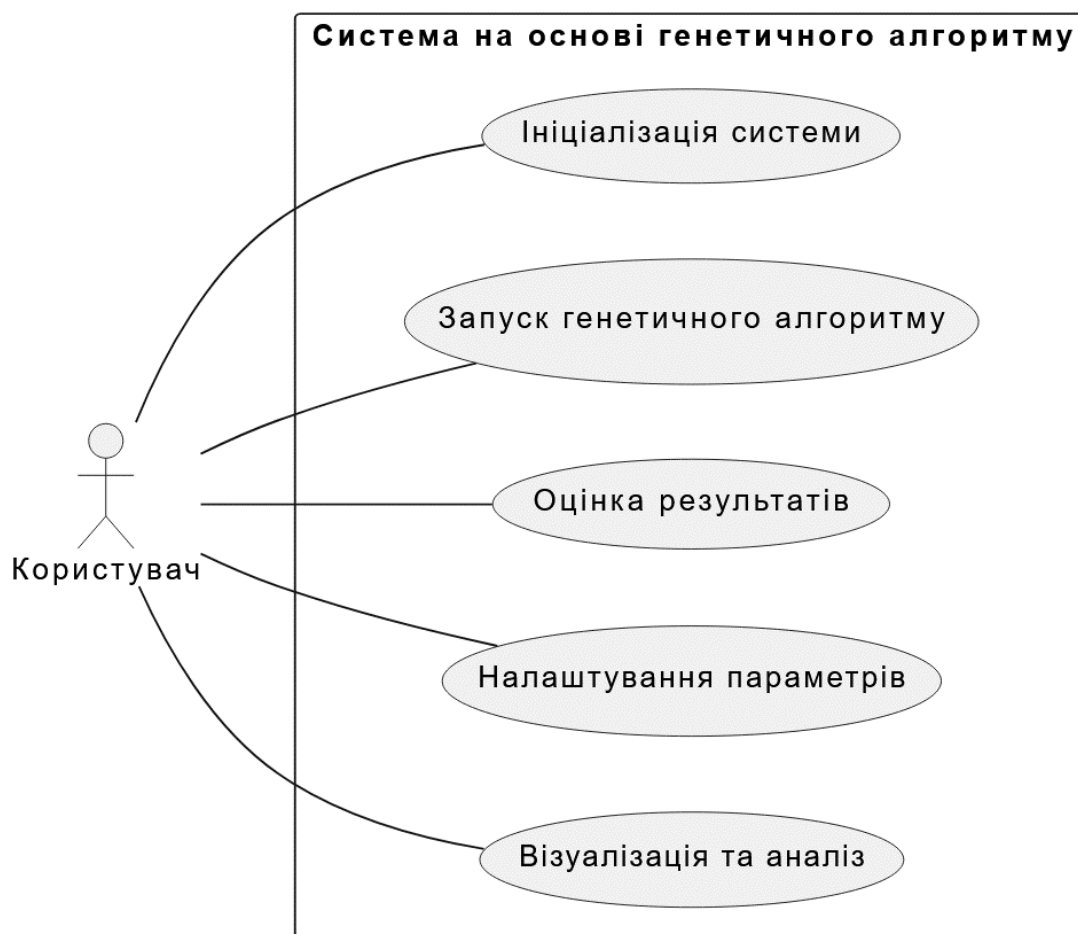


Рисунок 4.2 – Діаграма варіантів використання програми

Окрім цього було використано інструмент DEAP для підвищення ефективності та швидкодії навчання ГА. Хромосоми в ГА, які представляють потенційні рішення, моделюються як списки індексів, кожен з яких відповідає члену команди, призначеному для виконання певного завдання. Бібліотека DEAP спрощує створення, еволюцію та управління цими хромосомами, дозволяючи досліджувати різноманітну популяцію потенційних рішень.

Для забезпечення розуміння продуктивності та ефективності алгоритму реалізовано візуалізацію за допомогою таких інструментів, як Matplotlib та Seaborn.

Ці інструменти візуалізації забезпечують графічне представлення прогресу алгоритму в часі, проливаючи світло на його ефективність і поведінку. Такі візуалізації є не лише інформативними, але й критично важливими для точного налаштування параметрів алгоритму для оптимальної роботи.

Крім того, адаптивність і надійність системи підвищується завдяки використанню передових методів обробки даних. Використання Pandas DataFrames для організації та управління даними про завдання і членів команди підкреслює цей аспект. Таке структурування даних не тільки полегшує початкове налаштування генетичного алгоритму, але й відіграє вирішальну роль у його подальшій роботі.

Структури DataFrame забезпечують швидкий і ефективний доступ до пов'язаних із завданням атрибутів і характеристик членів команди, які необхідні для обчислення функції пристосованості. Такий рівень організації та управління даними гарантує, що генетичний алгоритм працює на основі точної та добре структурованої інформації, що призводить до більш надійних та ефективних командних завдань.

Таким чином, дизайн системи втілює гармонійне поєднання передових алгоритмічних підходів зі складними методами управління даними, кульмінацією якого є інструмент, що значно підвищує ефективність управління проектними командами.

Запропонована інтелектуальна технологія планування робіт використовує можливості генетичних алгоритмів для оптимізації розподілу завдань між командами для виконання проектних завдань.

Ця технологія покликана значно підвищити ефективність управління проектами шляхом інтелектуального узгодження навичок і доступності членів команди з вимогами проекту.

Архітектура системи, як показано на діаграмі діяльності (рис 4.3) ілюструє ітеративний і динамічний характер генетичного алгоритму, який є основою цієї інтелектуальної технології для планування.

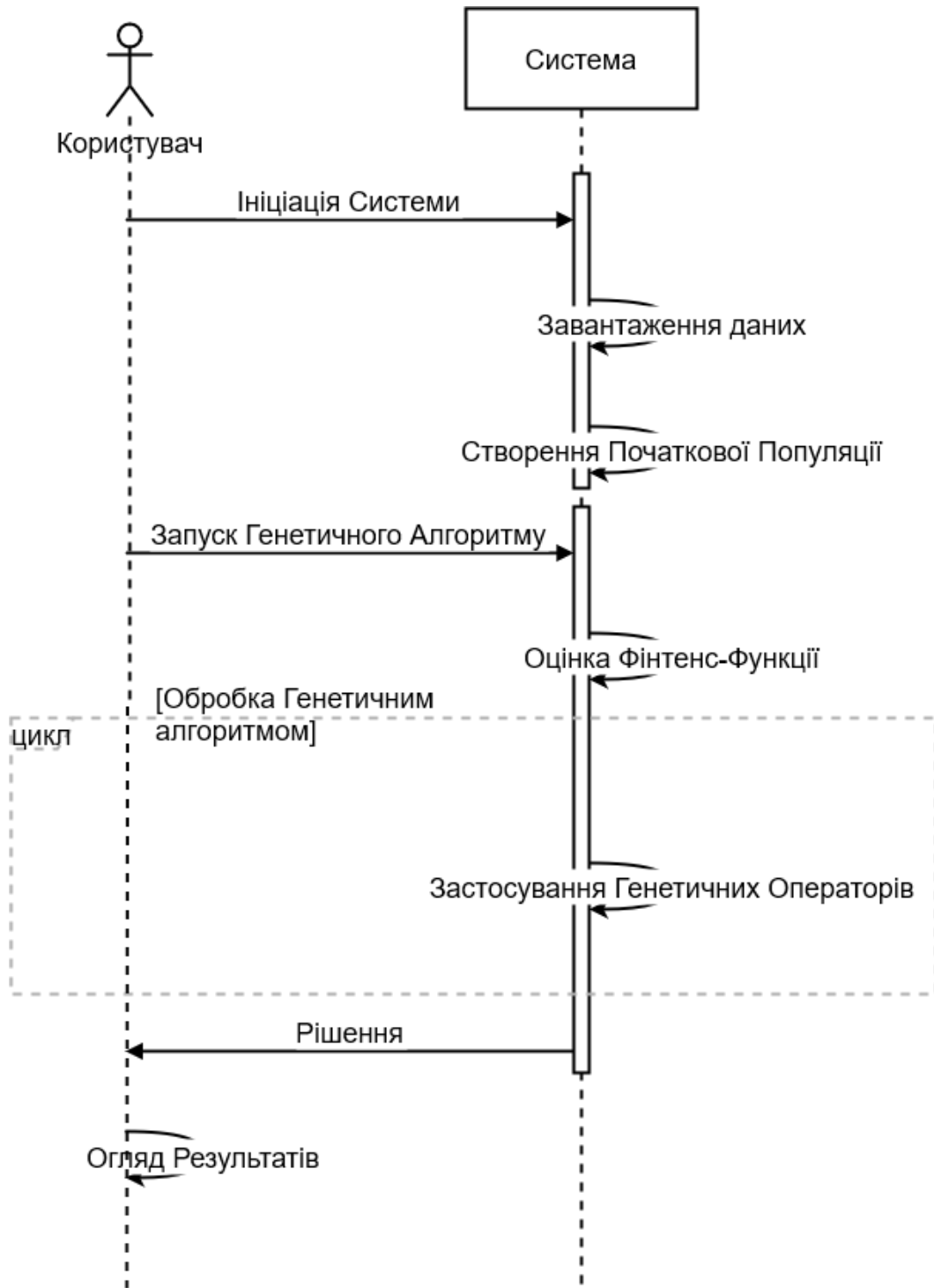


Рисунок 4.3 – Діаграма послідовності зв'язку користувача та системи

Процес починається з того, що користувач ініціалізує систему, яка потім завантажує завдання проєкту і доступних членів команди з шару даних. Генетичний алгоритм починається зі створення початкової популяції потенційних рішень. Кожна особа в популяції представляє унікальний розподіл членів команди на завдання. Потім ГА ітеративно оцінює, відбирає та

еволюціонує популяцію, використовуючи кросинговер і мутації, керуючись функцією пристосованості. Після досягнення критерію припинення, як правило, певної кількості поколінь, система виводить оптимальний розподіл команд. Користувач може переглянути результати і внести необхідні корективи або перезапустити алгоритм, якщо це необхідно.

В основі алгоритму лежить ініціалізація структур даних, що представляють завдання та членів команди. Це досягається за допомогою Pandas DataFrames, що пропонує універсальні та інтуїтивно зрозумілі засоби для роботи з даними.

```
tasks = pd.DataFrame({ ... }) # Дані, що стосуються завдань
team_members = pd.DataFrame({ ... }) # Дані про членів команди
```

У цих фреймах даних кожна задача і член команди інкапсульовані з відповідними атрибутами, такими як навички і доступність. Такий спосіб організації даних забезпечує легкий доступ до них та маніпулювання ними під час виконання алгоритму.

Починаючи зі стартового стану, система ініціалізує популяцію. Основний цикл включає оцінку придатності кожної особини, відбір найбільш придатних особин для розмноження та застосування кросинговеру і мутацій для створення нових особин.

Цей цикл триває до тих пір, поки не буде виконана умова припинення, що призводить до кінцевого стану, в якому представлено найкраще рішення. На діаграмі виділено точки прийняття рішень, такі як оцінка пристосованості та перевірка припинення, що підкреслює динамічну природу процесу ГА.

Діаграма активності (Рис. 4.4) ілюструє циклічну та ітеративну природу генетичного алгоритму в системі.

Критично важливим аспектом ГА є спосіб представлення рішень (хромосом). У цій реалізації хромосома - це список індексів, кожен з яких відповідає члену команди, призначеному на певну задачу.



Рисунок 4.4. Діаграма активності

```

creator.create("Individual", list, fitness=creator.FitnessMax)
toolbox.register("individual", tools.initlterate, creator.Individual,
create_individual)

```

```

population = toolbox.population(n=50)

```

Бібліотека DEAP полегшує створення та керування цими хромосомами, дозволяючи ініціалізувати різноманітну популяцію потенційних рішень.

Проведено тестування цих моделей, з використанням НД, які підібрані для нього, тобто тестові та валідаційні дані. Проте тестування показало, що основний метод є ефективнішим та точнішим, порівняно з експериментальним.

Також була створена матриця невідповідностей, що дала можливість точно показати рух нейронів в мережі, та як формувався вибір, чи це фейк чи реальне відео.

Функція придатності лежить в основі ГА, оцінюючи придатність кожного рішення. Вона враховує відповідність між навичками членів команди та вимогами завдань, а також їхню доступність.

```
def evaluate(individual):
    # Логіка оцінки фітнесу
    return total_score,
toolbox.register("evaluate", evaluate)
```

Ця функція кількісно оцінює кожну хромосому, спрямовуючи еволюційний процес до більш оптимальних рішень. Генетичні оператори дозволяють алгоритму досліджувати простір розв'язків і вдосконалювати рішення протягом поколінь. DEAP надає набір методів для реалізації цих операторів.

```
toolbox.register("mate", tools.cxUniform, indpb=0.5)
toolbox.register("mutate", tools.mutUniformInt, low=0,
up=len(team_members) - 1, indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)
```

Методи кросинговеру і мутації вносять у популяцію мінливість і нові ознаки, а метод відбору забезпечує відбір більш придатних особин для наступного покоління. Виконання ГА включає в себе ітерацію поколінь, застосування генетичних операторів та вибір найкращих рішень.

```
for gen in range(NGEN):
    # Операції генетичного алгоритму
```

Цей цикл є динамічною фазою, де популяція еволюціонує, керуючись функцією пристосованості та формуючись генетичними операторами. За допомогою Matplotlib та Seaborn візуалізовано роботу алгоритму, що дає уявлення про його ефективність.

```
plt.figure(figsize=(10, 5))
sns.lineplot(range(len(fitnesses)), fitnesses)
```

`plt.show()`

Ці візуалізації дають наочне уявлення про хід роботи алгоритму і мають вирішальне значення для аналізу його поведінки в часі. Система починається з ретельної організації завдань і членів команди в Pandas DataFrames. Цей крок - не просто зберігання даних, він включає попередню обробку та структурування даних для приведення їх у відповідність до вимог ГА. Наприклад, навички зберігаються у вигляді наборів для ефективних операцій підбору, а тривалість ретельно обробляється, щоб врахувати часові обмеження при плануванні проєкту.

Ефективність системи планування проєктів та формування команд на основі генетичних алгоритмів була оцінена за допомогою серії ретельно спланованих експериментів та аналізів. Цей процес оцінки має вирішальне значення для визначення не лише ефективності алгоритму у створенні оптимального складу команди, але й його надійності, ефективності та практичної застосовності в реальних умовах.

Експериментальна установка була ретельно розроблена, щоб імітувати різноманітні середовища планування проєктів. Варіації розміру популяції, частоти мутацій і кросинговеру, а також кількості поколінь дозволили отримати повне уявлення про поведінку системи за різних умов. В експериментах використовувалися різні проєктні завдання та члени команди, кожен з яких мав унікальні навички та доступність, що імітувало складнощі та виклики, з якими зазвичай стикаються в управлінні проєктами.

Аналіз результатів цих експериментів дозволив зробити кілька ключових висновків щодо ефективності системи. Якість рішень, виміряна за допомогою балів придатності, постійно вказувала на високий ступінь відповідності між навичками призначених членів команди та вимогами завдань. Ця відповідність має вирішальне значення для ефективного виконання проєкту і була основним показником спроможності системи.

Швидкість, з якою алгоритм збігався до стабільного розв'язку, варіювалася залежно від різних налаштувань, надаючи цінні дані про ефективність алгоритму.

Швидша збіжність була відзначена у випадках з більшими розмірами популяцій і вищими темпами мутацій, що свідчить про більш ретельне дослідження простору розв'язків.

Однак це також підкреслило компроміс між дослідженням і експлуатацією - баланс, який є ключовим для ефективності генетичних алгоритмів.

Стійкість і надійність системи оцінювалися за стабільністю її продуктивності під час численних запусків з різними початковими умовами і параметрами. Ця узгодженість є життєво важливою для забезпечення надійності системи і її надійного використання в різних сценаріях планування проєктів.

Практичність і зручність використання, важливі для застосування в реальних умовах, також були в центрі уваги при оцінці. Було перевірено адаптивність системи до різних проєктних вимог та її інтеграцію в існуючі процеси управління проєктами.

Гнучкість генетичного алгоритму в поєднанні з його здатністю вирішувати складні, багатогранні завдання планування проєктів підкреслили його практичну цінність.

Візуальне представлення даних у вигляді діаграм і графіків відіграло ключову роль в інтерпретації результатів. Вони не тільки зробили складні дані більш доступними, але й дозволили чіткіше зрозуміти процес оптимізації алгоритму та його продуктивність за різних параметрів.

На завершення, комплексна оцінка системи підкреслила її потенціал як високоефективного інструменту для планування проєктів та формування команд.

Здатність генетичного алгоритму генерувати оптимальні рішення в поєднанні з його адаптивністю та практичним застосуванням робить його цінним надбанням у сфері управління проєктами.

Хоча система продемонструвала значні сильні сторони, отримані знання також прокладають шлях для майбутніх удосконалень, зокрема, для оптимізації швидкості збіжності та подальшого підвищення її стійкості та надійності.

4.3 Вимоги до інтелектуалізованого планувальника роботи команди розробників проєктів

Для успішного виконання поставлених завдань було складено вимоги до інтелектуалізованого планувальника роботи команди розробників проєктів.

У відповідності до стандарту ISO/IEC 25010:2011, модель якості продукту розподіляє характеристики якості на вісім основних груп, кожна з яких містить декілька взаємопов'язаних підкатегорій:

- функціональність (включаючи повноту, точність та відповідність функцій);
- уфективність (оцінка часу відгуку, використання ресурсів та обсягу);
- сумісність (здатність співпрацювати та інтегруватися з іншими системами);
- зручність використання (включаючи інтуїтивність, навчальні можливості, продуктивність, запобігання помилкам користувача, естетику інтерфейсу та доступність);
- надійність (зрілість, доступність, стійкість до збоїв, можливість відновлення);
- безпека (конфіденційність, цілісність, незаперечність, відповідальність, автентичність);
- здатність до підтримки (модульність, можливість повторного використання, аналізування, змінюваність, перевіряємість);
- переносимість (адаптивність, встановлюваність, замінюваність).

Відповідно до цих характеристик, у відповідності до стандарту ISO/IEC 25010:2011, були розроблені вимоги до технологій ідентифікації решти дефектів у програмному забезпеченні (таблиця 4.1).

Таблиця 4.1 – Вимоги до інформаційної технології ідентифікації залишкових дефектів у ПЗ

Категорія вимог	Опис
Загальні вимоги	Система має здатність аналізувати код та використовувати тестові сценарії для виявлення та класифікації залишкових помилок у програмному забезпеченні автоматично.
Функціональність	Автоматичне виявлення та класифікація помилок на всіх стадіях розробки. Встановлення порогів серйозності помилок та інтерфейс для аналізу виявлених помилок.
Продуктивність	Швидка обробка великої кількості коду, з встановленням максимального часу на виявлення помилок.
Надійність та безпека	Захист від несанкціонованого доступу та забезпечення стабільної роботи системи.
Інтеграція та сумісність	Легке інтегрування з існуючими інструментами розробки та тестування, а також сумісність з різними платформами.
Моніторинг та звітність	Функції ведення журналу виявлених помилок та статусів, а також генерація звітів.
Автоматизація	Підтримка автоматизованої ідентифікації та аналізу коду, можливість налаштування автоматичних сповіщень про нові помилки.
Адаптивність	Гнучкість системи для підтримки різних типів програмного забезпечення, мов програмування та архітектурних підходів.

Продовження таблиці 4.1 – Вимоги до інформаційної технології ідентифікації залишкових дефектів у ПЗ

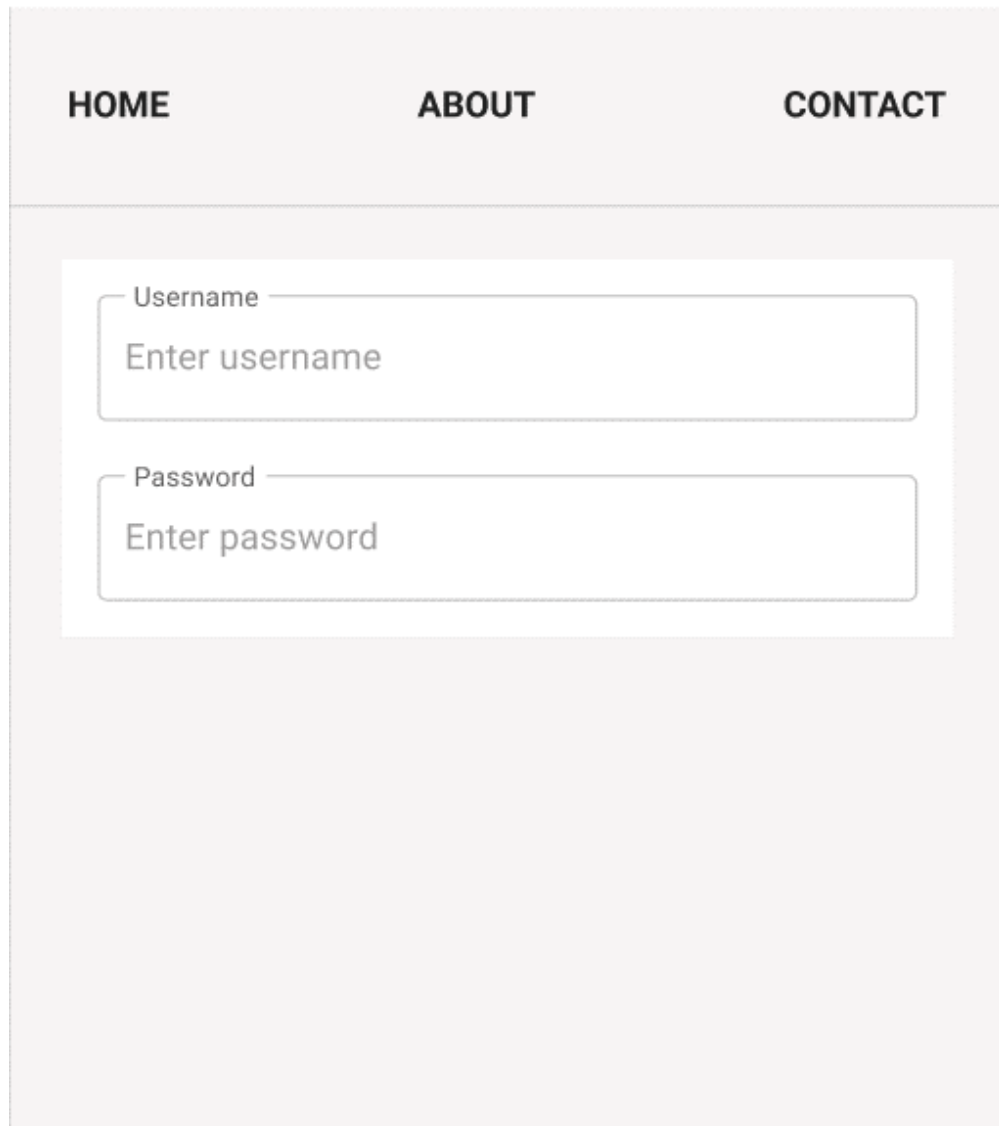
Категорія вимог	Опис
Зручність використання	Інтуїтивно зрозумілий інтерфейс з можливостями фільтрації, сортування та пошуку.
Ресурсне споживання	Мінімізація впливу на продуктивність користувачів під час роботи з системою.
Доступність	Забезпечення доступу до системи для користувачів із різними потребами доступності.
Сумісність із стандартами	Дотримання існуючих стандартів безпеки для обробки та зберігання конфіденційних даних.
Конфіденційність	Заходи для захисту конфіденційних даних, таких як результати ідентифікації дефектів.
Тестування	Встановлення вимог до тестування інформаційної технології, включаючи критерії ефективності.
Управління життєвим циклом	Підтримка оновлення та підтримки системи протягом всього життєвого циклу програмного забезпечення.
Документація	Надання докладної документації для використання, налаштування та адміністрування системи.

4.4 Огляд розробленого інтерфейсу додатку для інтелектуалізованого планувальника роботи команди розробників проєктів

Задля проведення оцінки ефективності розробленого алгоритму було створено програмне забезпечення.

Інтерфейс автентифікації є безпечною точкою входу до інтелектуальної системи планування. Цей портал слугує основним рівнем захисту, захищаючи дані проєкту від несанкціонованого доступу.

Дизайн інтерфейсу, який включає окремі поля для введення імені користувача і пароля, відповідає сучасним нормам безпеки, рисунок 4.5.



The image shows a user authentication interface. At the top, there are three navigation links: HOME, ABOUT, and CONTACT. Below them is a white rectangular form containing two input fields. The first field is labeled 'Username' and contains the placeholder text 'Enter username'. The second field is labeled 'Password' and contains the placeholder text 'Enter password'.

Рисунок 4.5 – Інтерфейс аутентифікації

Інтерфейс налаштувань користувача є елементом інтелектуального планувальника, який дозволяє окремим користувачам налаштовувати програму відповідно до своїх вимог.

Опція вибору теми пропонує персоналізоване візуальне середовище, яке може допомогти зменшити когнітивне навантаження і підвищити концентрацію користувача.

Крім того це вікно також містить опції конфігурації системи, які дозволяють користувачам перемикаати сповіщення, демонструють розуміння важливості автономії користувача в управлінні інформацією.

Цей інтерфейс втілює принципи дизайну, орієнтованого на користувача, що робить планувальник не тільки інструментом управління проектами, але й персоналізованим помічником, який задовольняє різні уподобання кожного члена команди, рисунок 4.6.

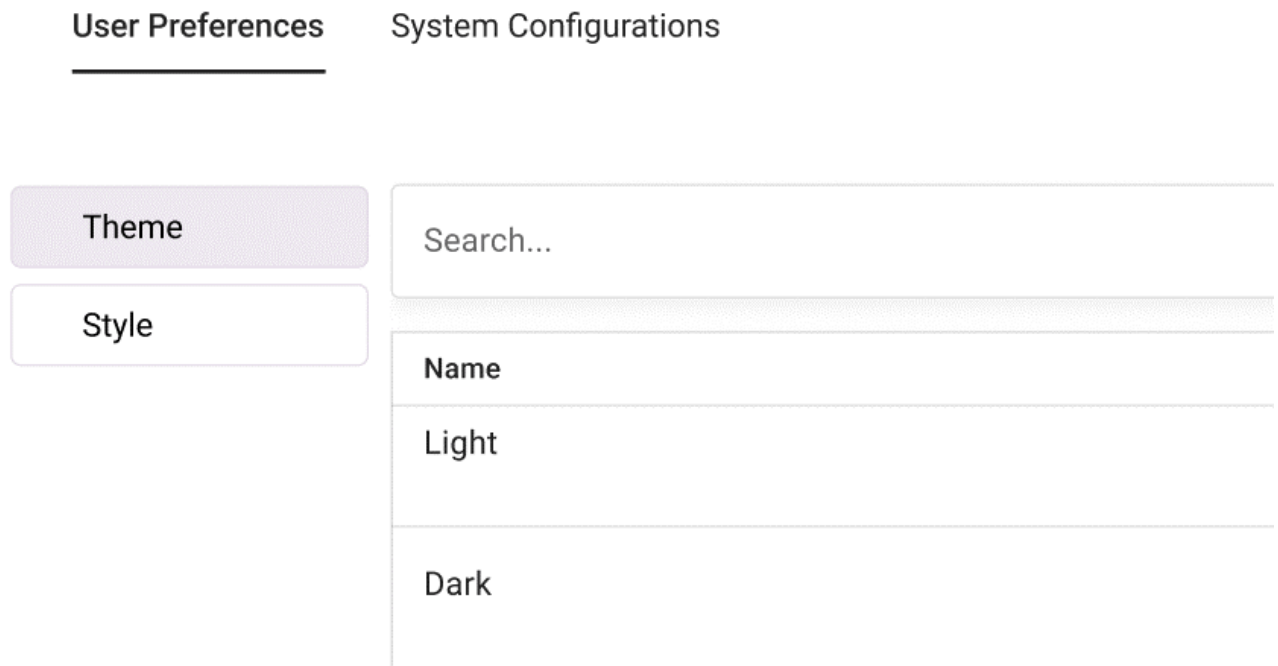


Рисунок 4.6 – Інтерфейс налаштувань користувача

Інтерфейс Dashboard представляє огляд поточного стану проекту, що відображає центральну роль планувальника у відстеженні та звітності.

Кількість незавершених завдань слугує безпосереднім індикатором завантаженості проекту, в той час як кількість активних членів команди дає уявлення про використання команди (рисунок 4.7).

Цей модуль відображає відповідні метрики, такі як кількість незавершених завдань і активних членів команди, які мають вирішальне значення для наглядових функцій менеджера проекту.

Цей інтерфейс візуально представляє результати роботи алгоритму, що дозволяє спростити перегляд складних проектних даних. Цей інтерфейс підтверджує акцент на ефективному розподілі людських ресурсів.

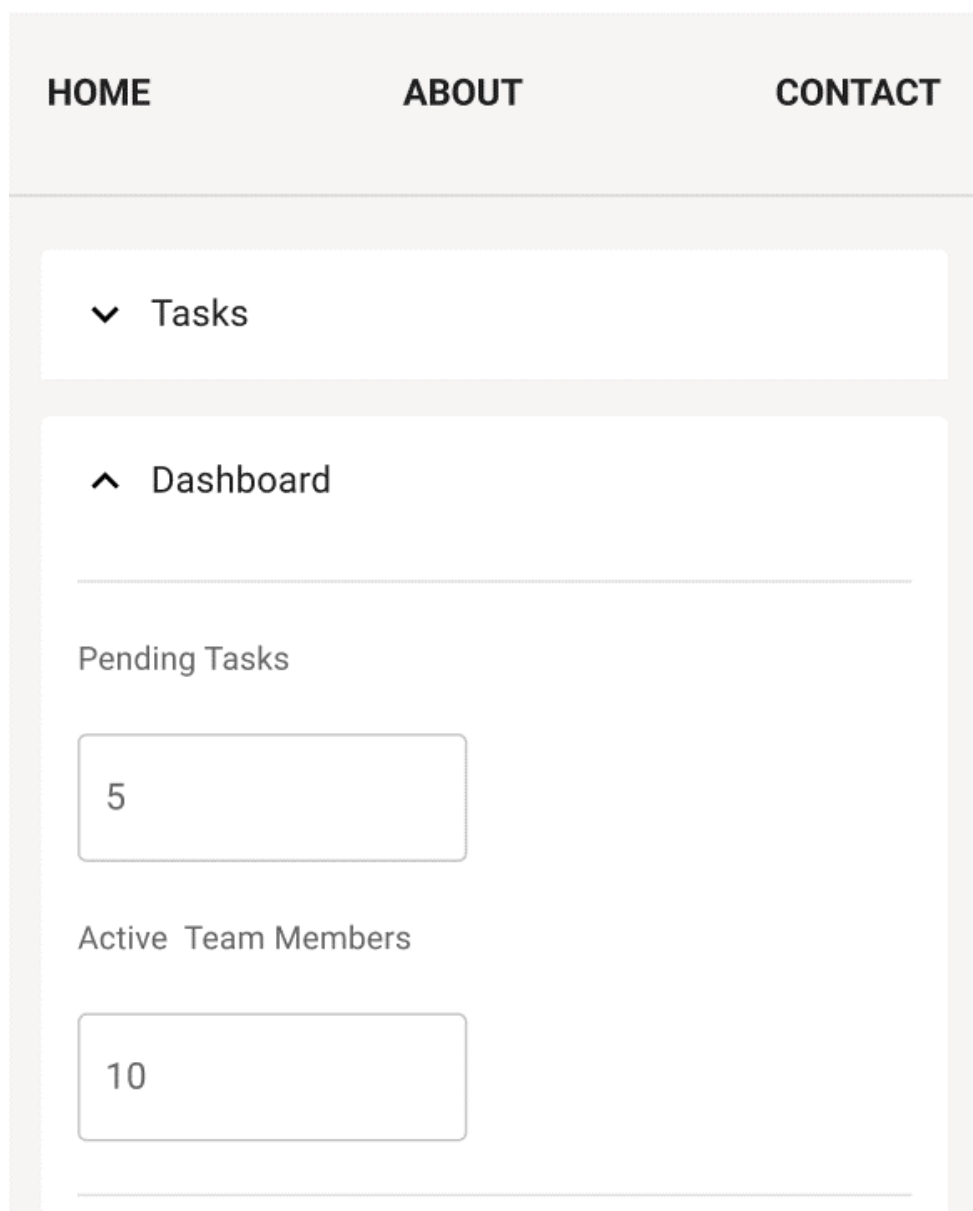


Рисунок 4.7 – Інтерфейс Dashboard

Інтерфейс керування завданнями втілює практичні аспекти функціональності планувальника. Він представляє завдання і відповідні їм терміни, а також надає можливість редагування або видалення, що відображає динамічну природу управління проектами.

Цей інтерфейс має фундаментальне значення для менеджерів проектів, щоб контролювати просування і вносити відповідні зміни, гарантуючи, що проект не відхиляється від курсу і ресурси розподіляються належним чином (рисунок .4.8)

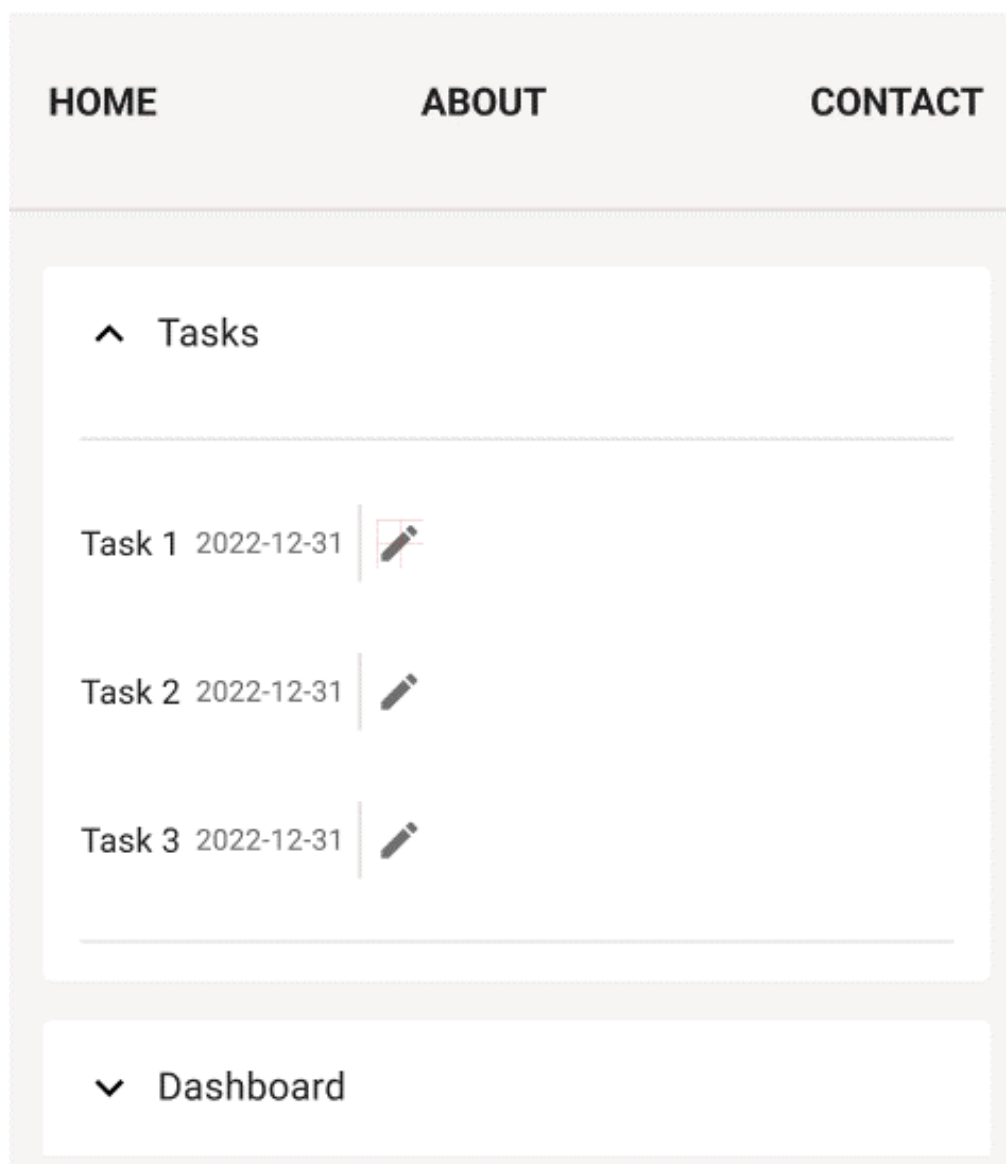


Рисунок 4.8 –Інтерфейс керування завданнями

Цей користувацький інтерфейс є практичною реалізацією теоретичної концепції статті, наочно і функціонально ілюструючи роль генетичного алгоритму в призначенні завдань. Представлення назви завдання разом з дедлайном відображає логіку планувальника, який оцінює навички членів команди відповідно до вимог завдання, а також елементи завершення завдання, що базуються на часі.

4.5 Висновки

У розділі була представлена інформаційна технологія інтелектуалізованого планувальника роботи команди розробників проєктів, включаючи детальний огляд її архітектури та визначення основних вимог. Було також описано інтерфейс розробленого додатку, який спрощує взаємодію користувачів з системою планування. Основною метою цієї технології є підвищення ефективності управління проєктами за допомогою автоматизації процесів розподілу завдань та ресурсів, враховуючи динамічну природу проєктної роботи.

Технологія базується на застосуванні розширених алгоритмів та інноваційних підходів у плануванні, забезпечуючи високу точність та адаптивність до потреб команди. Результати тестування інтерфейсу та функціональності інтелектуалізованого планувальника показали, що система є ефективним інструментом для поліпшення управління проєктами, забезпечуючи високий рівень точності та зручності використання.

Проведені дослідження та експерименти виявили, що розроблена технологія має високу ефективність у вирішенні складних завдань планування, при цьому демонструючи гнучкість у врахуванні різноманітних аспектів проєктної діяльності та потреб команди розробників.

ВИСНОВКИ

У кваліфікаційній роботі було проведено ряд теоретичних та практичних досліджень, присвячених удосконаленню метода, а також реалізації інтелектуалізованого планування роботи команд розробників проектів.

У першому розділі було розглянуто існуючі проблеми та виклики у сфері планування роботи команд розробників, а також представлено основні підходи до розв'язання цих проблем.

У другому розділі було зосереджено увагу на вивченні та аналізі генетичних алгоритмів, їх застосування для оптимізації планування проектів та формування команд. Розроблено підходи до обробки та аналізу даних, що дозволяють підвищити ефективність планування.

Третій розділ охоплює розробку та впровадження удосконаленого методу планування роботи, заснованого на генетичних алгоритмах. Було продемонстровано, як цей метод може бути застосований для вирішення конкретних завдань планування та управління проектами, забезпечуючи оптимальний розподіл завдань та ресурсів серед членів команди. У четвертому розділі була представлена детальна архітектура та функціональні вимоги до інтелектуалізованого планувальника роботи.

Огляд розробленого інтерфейсу додатку демонструє його зручність та ефективність у взаємодії з користувачами, що сприяє підвищенню продуктивності команди розробників. Виконані експериментальні дослідження підтверджують високу ефективність та надійність запропонованого методу.

Загалом, результати цієї роботи відкривають нові перспективи для управління командами розробників проектів, пропонуючи комплексний та інноваційний підхід до планування та оптимізації процесів. Розроблений метод та інструменти є значним внеском у сферу управління проектами, демонструючи потенціал для подальших досліджень та розвитку.

На основі експериментів було продемонстровано що розроблена технологія показує приріст у часі збіжності функції у порівнянні з відомими методами

середньому на 3%. Коли мова заходить за масштабування то розроблений метод показує приріст на 4% порівняно з існуючими технологіями, що були продемонстровані в роботі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Мандрик А., Лисенко С. Метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму. *Актуальні проблеми комп'ютерних наук АПКН-2023*: зб. праць XV Всеукраїнської науково-практичної конференції, Хмельницький, м.Хмельницький, ХНУ, 17-18 листопада 2023. С. 177–183.
2. Мандрик А., Лисенко С. Метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму. *Вісник Хмельницького Національного Університету*. 2023.
3. Standish Group CHAOS Report. Standish Group International. URL: <https://hennyportman.files.wordpress.com/2021/01/project-success-qrc-standish-group-chaos-report-2020.pdf> (дата звернення: 20/09/2023).
4. José G. M. Esgario, Iago E. da Silva, and Renato A. Krohling. Application of Genetic Algorithms to the Multiple Team Formation Problem. *Arxiv*, 2019. <https://doi.org/10.48550/arXiv.1903.03523>
5. Mahendra G. All About Project Scheduling. URL: <https://www.saviom.com/blog/what-is-project-scheduling-and-why-is-it-important/> (дата звернення: 22/11/2023).
6. Mahfouz R., Neil H., and Rasheed E. The Scheduling of Problems. *Journal of Combinatorial Theory*. 2020. vol. 139. Pp. 59-79. <https://doi.org/10.1016/j.jcta.2015.11.001>
7. Y.-B. Hu, Y.-P. Wang, F.-Y. Guo. A New Penalty based Genetic Algorithm for Constrained Optimization Problems. *International Conference on Machine Learning and Cybernetics*. 2017. vol. 5. pp. 3025–3029.
8. Miller B. L, Goldberg D. E. Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems*. 2019. vol. 9. pp. 193–212
9. Appleby, D., Watabe, A., Gilby, S. School Project Planning Process. 2021. 20–23 pp.

10. Ibn M., Abdellah B., Abdelouahhab J. A Novel Approach for Integrating the Optimization of the Lifetime and Cost of Manufacturing of a New Product during the Design Phase. 2021. 345p.
11. Ibn M., Abdellah B., Abdelouahhab J. A Comparative Analysis of Metaheuristic Approaches (Genetic Algorithm/Hybridization of Genetic Algorithms and Simulated Annealing) for Planning and Scheduling Problem with Energy Aspect. 2021. 363p.
12. Twedle J., Roy, H., Robinson, L., Pock, M. J., Noss, R. . 2020. 33–50pp.
13. Beratan K. Improving Problem Definition and Project Planning in Complex Natural Resource Management Problem Situations Using Knowledge Brokers and Visual Design Principles. *Ecology and Society*. 2019. vol.24., pp. 10-15.
14. Jason H. Project Management: From Idea to Implementation. 2022. 79–92pp.
15. Jordan M., Feild H. S. The Relationship of Group Process Variables and Team Performance: A Team-level Analysis in a Field Setting. *Small Group Research*. 2022. vol. 33, no. 1. 121–150 pp.
16. Beasley D., Bull, D., and Martin, R. A Sequential Niche Technique for Multimodal Function Optimization. *Evolutionary Computation*. 2020. pp. 101–125
17. Chang C., Jiang H., Di Y., Zhu D. Time-Line Based Model for Software Project Scheduling with Genetic Algorithms. *Inform. Software Tech*. 2018. pp. 1142-1154
18. Chang K., Christensen, M.J., and Zhang, T. Genetic Algorithms for Project Management. 2021. 107–139 pp.
19. Pan N., Hsaio P., Chen K. A Study of Project Scheduling Optimization using TabuSearch Algorithm. 2020. 1101–1112 pp.
20. Introduction to Project Schedule Management. URL: <https://www.migso-rcubed.com/blog/schedule-management/> (дата звернення: 11/09/2023).
21. Xia M., Li Y., Shen Y. Loading Sequencing Problem in Container Terminal with Deep Q-Learning. *Journal of Coastal Research*. 2020. vol. 103 pp. 817–821

22. Joslin D., and Poole W. Agent-based Simulation for Software Project Planning. *37th Winter Simulation Conference*. 2021. 1059–1066 pp.
23. Acuña S., Juristo N., Moreno M. A Software Process Model. *Handbook for Incorporating People's Capabilities*. New Yor. 2019.
24. Edirlei S., Bruno F. Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning. *SBGames*. 2019.
25. Mirjalili Y., Seyedali K., Seyedali M. Genetic algorithm. Evolutionary Algorithms and Neural Networks. *Theory and Applications*. 2019. vol. 7 43-55pp.
26. Mitsuo P., Lin L. Genetic algorithms and their applications. *Springer handbook of engineering statistics*. London. 2023. pp. 635-674
27. Lee K. A review of applications of genetic algorithms in operations management. *Engineering Applications of Artificial Intelligence*. 2018. 1-12 pp.
28. Sohail A. Genetic algorithms in the fields of artificial intelligence and data sciences. *Annals of Data Science*. 2023. 1007-1018 pp.
29. Hassanat K., Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach *Information Tech*. 2019. vol. 6, p. 390.
30. Chekol T., Nuramo A. Key Factors Contributing to Time and Cost Overrun in Mega Sugar Construction Projects in Ethiopia. *Leuven University Press* . 2020. pp. 473–498.
31. Katoch S., Chauhan, S., Kumar, V. A review on genetic algorithm: past, present, and future. 2021.
32. Yongping L., Lizhen H., Xiufeng L., Guomin J., Xu, C., Erling, O. A late-mover algorithm for resource-constrained project-scheduling problems. *Information Sciences*. 2023. vol. 642. pp. 20-255
33. Wang N., and Wei L. Mathematical Model of Port Scheduling Problem Based on Multi-Objective Particle Swarm Optimization. *Journal of Coastal Research*. 2020. pp. 561–65.
34. Javadi A. A hybrid intelligent genetic algorithm. *Advanced Engineering Informatics*. 2018. pp. 255-262.

35. Farooq M. Genetic algorithm technique in hybrid intelligent systems for pattern recognition. *International Journal of Innovative Research in Science, Engineering and Technology*. 2018. pp. 1891-1898.
36. Amiri A. Comparison between artificial neural network and hybrid intelligent genetic algorithm in predicting the severity of fixed object crashes among elderly drivers. 2020. 138 P.
37. Renard S. From Incubation to Delivery: An Application of Project Management for the Music Industry. 2020. 1–18 pp.
38. Bhatawdekar R. The effects of particle swarm optimisation and genetic algorithm on ANN results in predicting pile bearing capacity. *International Journal of Hydromechatronics*. 2020. pp. 69-87.
39. ProjectLibre for PC URL: <https://www.amazon.com/ProjectLibre-PC-Open-Source-Download/dp/B00PT6H73S/> (дата звернення: 17/11/2023).
40. Shu W., Ken C., Neal X. Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing. *Future Generation Computer Systems*. 2021. p. 20.
41. What is Project Scheduling, and Why is it Important? URL: <https://www.saviom.com/blog/what-is-project-scheduling-and-why-is-it-important/> (дата звернення: 22/11/2023).
42. Sharif N. Optimization methods for power scheduling problems in smart home: Survey. *Renewable and Sustainable Energy Reviews* . 2019. vol. 115.
43. Balouka N., Izack C. A robust optimization approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*. 2021. pp. 457-470
44. Martin F. A Genetic Algorithm for Personnel Scheduling in Vacation Seasons. 2022. URL: <https://www.diva-portal.org/smash/get/diva2:1641695/FULLTEXT01.pdf> (дата звернення: 21/10/2023).

45. A genetic algorithm scheduled our developer conference. Here's what I learnt. URL: <https://filiph.medium.com/a-genetic-algorithm-scheduled-our-developer-conference-heres-what-i-learnt-eac0069709f5> (дата звернення: 24/11/2023).
46. Lu S. Multi-Objective Workshop Scheduling of Marine Production Based on Improved Ant Colony Algorithm. 2020. 222 P.
47. X Cai. A genetic algorithm for scheduling staff of mixed skills under multi-criteria. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0377221799003914> (дата звернення: 18/09/2023).
48. Debora J., Sharon L. Time and Trouble Management. 2020. 33–46 pp.
49. Staff Scheduling by a Genetic Algorithm with a Two-Dimensional Chromosome Structure. URL: <https://www.patatconference.org/patat2008/proceedings/Dean-WA3c.pdf> (дата звернення: 20/09/2023).
50. Pellerin R., Perrier N. Review of methods, techniques and tools for project planning and control. *International Journal of Production Research*. 2019. vol. 57. pp. 2160-2178.
51. Bell J., Sharon L. , John C. Time and Trouble Management. *Psychology and Related Fields*, 3rd ed. 2022. pp. 33–46
52. Morariu C. Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. 2020. 1032 p.
53. Arunarani A., Manjula D., Sugumaran V. Task scheduling techniques in cloud computing. *A literature survey*. 2019. 407-415 pp.
54. Kumar M. Comprehensive explanation for scheduling techniques in cloud computing. 2019. pp. 1-33.
55. Morandini M. Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology. *A survey for analyzing results for development teams*. 2021. P. 100.

56. Hidayati A., Budiardjo E., Purwandari B. Hard and soft skills for scrum global software development teams. *Proceedings of the 3rd International Conference on Software Engineering and Information Management*. 2020. pp. 77-102.
57. Using Genetic Algorithms to Improve Prediction of Execution Times of ML Tasks. URL: https://link.springer.com/chapter/10.1007/978-3-642-28942-2_18 (дата звернення: 18/09/2023).
58. Hector Y., Nayeli Y., Melba J. , and Kenneth S. Pope. Finding the right software and system for scheduling. 2023. 51–58pp.
59. Li W., Risk Analysis Model of Offshore Engineering Project Management Based on Fuzzy Membership Function. 2019, 92–95 pp.
60. Optimizing LSTM Network using Genetic Algorithm for Stock Market Price Prediction. URL: <https://www.linkedin.com/pulse/optimizing-lstm-network-using-genetic-algorithm-stock-akash-dangi/> (дата звернення: 11/09/2023).
61. Genetic algorithm-based hyperparameter optimization of deep learning models for PM2.5 time-series prediction. URL: <https://link.springer.com/article/10.1007/s13762-023-04763-6> (дата звернення: 29/09/2023).
62. Genetic Algorithm for the Optimization of a Building Power Consumption Prediction Model. URL: <https://www.mdpi.com/2079-9292/11/21/3591> (дата звернення: 04/10/2023).
63. Project Scheduling 101 for 2023 URL: <https://www.mdpi.com/2079-9292/11/21/3591> (дата звернення: 05/11/2023).
64. Mitchell L. Scheduling and Staffing the Work. 2023. 79–104 pp.
65. Applications of Genetic Algorithms in Machine Learning. URL: <https://www.turing.com/kb/genetic-algorithm-applications-in-ml> (дата звернення: 17/09/2023).
66. Debora J., Sharon L. Foster, and John D. Cone. Time and Trouble Management. 2020. 33–46 pp.
67. Gallego J. , Ortiz-Marcos I., Romero J. Main challenges during project planning when working with virtual teams. 2021.

68. Bardakci G. Project Management: Assessment of Risk Factors to Cost. *Schedule in High-Rise Building Projects*. 2020. pp. 28–35
69. Rong P., Li C., Xie J. Learning, trust, and creativity in top management teams. *Team reflexivity as a moderator survey*. 2019. pp. 1-14
70. Cosma O., Pop P. A two-level based genetic algorithm for solving the soft-clustered vehicle routing problem. *Carpathian Journal of Mathematics* . 2021. vol. 38 pp.117–128. URL: <https://www.jstor.org/stable/27082123>
71. Wu X. Shared mental models in multi-team systems: improving enterprise system implementation. 2023. 185-208 pp.
72. Application of Genetic Algorithms for Decision-Making in Project Management: A Literature Review. URL: https://www.researchgate.net/publication/338125559_Application_of_Genetic_Algorithms_for_Decision-Making_in_Project_Management_A_Literature_Review (дата звернення: 22/10/2023).
73. Optimizing the Self-Organizing Team Size Using a Genetic Algorithm in Agile Practices. URL: <https://www.degruyter.com/document/doi/10.1515/jisys-2018-0085/html?lang=en> (дата звернення: 14/10/2023).
74. Li M., Qin S. Study on Generation Scheduling of Cascade Hydropower Stations Based on SAPSO. 2020.
75. Genetic Algorithms - Application Areas. URL: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_application_areas.htm (дата звернення: 16/10/2023).
76. Marion T. , Fixson, S. The transformation of the innovation process: How digital tools are changing work, collaboration, and organizations in new product development. 2021. 192-215 pp.
77. Čeh I., Štorga M., Delač G. Agent-Based Modelling: Parallel and Distributed Simulation of Product Development Team. 2022. 1424-1432 pp.
78. Leonori S. Optimization strategies for Microgrid energy management systems by Genetic Algorithms. 2020. 10 P.

79. Pan L. Predicting compressive strength of green concrete using hybrid artificial neural network with genetic algorithm. 2023. 1980-1996 pp.

80. Shift Management and Staff Scheduling. URL: <https://www.microsoft.com/en-gb/microsoft-teams/staff-scheduling-shift-management> (дата звернення: 21/10/2023).

81. Brewer J., Dittman K. Project Schedule and Cost Planning. *Methods of IT Project Management*. 4th ed. 2023. pp. 169–218.

82. The 8 best meeting scheduler apps in 2023. URL: <https://zapier.com/blog/best-meeting-scheduler-apps/> (дата звернення: 03/10/2023).

83. 3 Best Employee Scheduling Techniques to Save Your Time. URL: <https://www.timechamp.io/3-Employee-Scheduling-Techniques-to-Save-Your-Time-and-Sanity.html> (дата звернення: 20/10/2023).

ДОДАТОК А

ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНТЕЛЕКТУАЛІЗОВАНОГО ПЛАНУВАЛЬНИКА РОБОТИ КОМАНДИ РОЗРОБНИКІВ ПРОЄКТІВ

```

from app import db

class TeamMember(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(64), index=True, unique=True)
    skills = db.Column(db.String(120))
    availability = db.Column(db.Integer)
    tasks = db.relationship('Task', backref='team_member', lazy='dynamic')

    def __repr__(self):
        return '<TeamMember {}>'.format(self.name)
from app import db

class TeamMember(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(64), index=True, unique=True)
    skills = db.Column(db.String(120))
    availability = db.Column(db.Integer)
    tasks = db.relationship('Task', backref='team_member', lazy='dynamic')

    def __repr__(self):
        return '<TeamMember {}>'.format(self.name)
from app import db

class Task(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    description = db.Column(db.String(120))
    duration = db.Column(db.Integer)
    skills_required = db.Column(db.String(120))
    dependencies = db.Column(db.String(120))
    project_id = db.Column(db.Integer, db.ForeignKey('project.id'))

    # Foreign Key - ensures a task is associated with a team member
    team_member_id = db.Column(db.Integer, db.ForeignKey('team_member.id'))

    def __repr__(self):
        return '<Task {}>'.format(self.description)
from deap import base,
creator, tools, algorithms
from app.models.task import Task
from app.models.team_member import TeamMember

def schedule_tasks():
    # Define the fitness function and individual
    creator.create("FitnessMax", base.Fitness, weights=(1.0,))
    creator.create("Individual", list, fitness=creator.FitnessMax)

    # Create the toolbox
    toolbox = base.Toolbox()

    # Define the gene and individual creation operations
    toolbox.register("attr_bool", random.randint, 0, 1)
    toolbox.register("individual", tools.initRepeat, creator.Individual,
toolbox.attr_bool, n=100)
    toolbox.register("population", tools.initRepeat, list, toolbox.individual)

```

```

# Define the evaluation function
def evalOneMax(individual):
    return sum(individual),

# Define the genetic operators
toolbox.register("evaluate", evalOneMax)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

# Create the initial population
pop = toolbox.population(n=300)

# Run the genetic algorithm
algorithms.eaSimple(pop, toolbox, cxpb=0.5, mutpb=0.2, ngen=40,
verbose=False)

# Return the best individual
return tools.selBest(pop, k=1)[0]
from deap import base, creator, tools
import random

class GeneticAlgorithm:
    def __init__(self, tasks, team_members):
        self.tasks = tasks
        self.team_members = team_members

        creator.create("FitnessMax", base.Fitness, weights=(1.0,))
        creator.create("Individual", list, fitness=creator.FitnessMax)

        self.toolbox = base.Toolbox()
        self.toolbox.register("attr_bool", random.randint, 0, 1)
        self.toolbox.register("individual", tools.initRepeat,
creator.Individual, self.toolbox.attr_bool, len(self.tasks))
        self.toolbox.register("population", tools.initRepeat, list,
self.toolbox.individual)

        self.toolbox.register("evaluate", self.evaluate)
        self.toolbox.register("mate", tools.cxTwoPoint)
        self.toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
        self.toolbox.register("select", tools.selTournament, tournsize=3)

    def evaluate(self, individual):
        # This is a simplified evaluation function
        return sum(individual),

    def run(self, ngen=10):
        pop = self.toolbox.population(n=50)
        fitnesses = list(map(self.toolbox.evaluate, pop))
        for ind, fit in zip(pop, fitnesses):
            ind.fitness.values = fit

        for g in range(ngen):
            offspring = self.toolbox.select(pop, len(pop))
            offspring = list(map(self.toolbox.clone, offspring))

            for child1, child2 in zip(offspring[::2], offspring[1::2]):
                if random.random() < 0.5:
                    self.toolbox.mate(child1, child2)
                    del child1.fitness.values
                    del child2.fitness.values

```

```

    for mutant in offspring:
        if random.random() < 0.2:
            self.toolbox.mutate(mutant)
            del mutant.fitness.values

    invalid_ind = [ind for ind in offspring if not ind.fitness.valid]
    fitnesses = map(self.toolbox.evaluate, invalid_ind)
    for ind, fit in zip(invalid_ind, fitnesses):
        ind.fitness.values = fit

    pop[:] = offspring
import pandas as pd

class Task:
    def __init__(self, id, description, complexity):
        self.id = id
        self.description = description
        self.complexity = complexity

    def to_dict(self):
        return {
            'id': self.id,
            'description': self.description,
            'complexity': self.complexity
        }

    @staticmethod
    def from_dict(dict):
        return Task(dict['id'], dict['description'], dict['complexity'])

    @staticmethod
    def create_dataframe(tasks):
        return pd.DataFrame([task.to_dict() for task in tasks])
import pandas as pd

class TeamMember:
    def __init__(self, id, name, skills, availability):
        self.id = id
        self.name = name
        self.skills = skills
        self.availability = availability

    def to_dict(self):
        return {
            'id': self.id,
            'name': self.name,
            'skills': self.skills,
            'availability': self.availability
        }

    @staticmethod
    def from_dict(dict):
        return TeamMember(dict['id'], dict['name'], dict['skills'],
dict['availability'])

    @staticmethod
    def create_dataframe(team_members):
        return pd.DataFrame([team_member.to_dict() for team_member in
team_members])

```

ДОДАТОК Б

КОПІЯ ОПУБЛІКОВАНОЇ НАУКОВОЇ СТАТТІ

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2023»

17-18 листопада 2023

Хмельницький 2023

Козенко О.В., Мазурець О.В., Молчанова М.О., Собко О.В. Використання метрик косинусної схожості та індексу Жаккара для інтелектуального аналізу семантичної подібності текстових документів	146
Комін А.С., Бойко О.В. Архітектурне рішення для підсистеми підтримки управління гібридною енергосистемою з використанням машинного навчання на мобільних пристроях	148
Кузьмін А.А. Концепція інформаційної системи для автоматизованої генерації цифрового контенту на основі штучного інтелекту	153
Кучменко К.Ю., Праворська Н.І. Ігровий застосунок у жанрі «платформер» з інтерфейсом управління на основі голосової взаємодії з використанням технологій Unity	157
Лаптев М.П., Лисий А.М., Сергєєв Є.В., Віжевський П.В. Метод криптографічного захисту протоколів в засобах комунікації інтернету речей	161
Левандовський А.О., Муляр І.В. Метод аналізу трафіку з метою виявлення атак на комплексні системи захисту інформації	163
Лигун О.О. Методи та засоби виявлення зловмисних дроперів в комп'ютерних системах	166
Мазур К.Р., Пасічник О.А., Скрипник Т.К. Метод виявлення боєприпасів, що не розірвались, за зображенням з тепловізора засобами глибокого навчання	168
Малицький Т.Б., Чешун О.В., Чешун В.М. Математична інтерпретація концепції захисту інформаційних ресурсів корпоративної мережі із застосуванням імовірнісних критеріїв довіри	172
Мандрик А.І., Лисенко С.М. Метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму	177
Манзюк Е.А. Застосування розпаралелювання для криптографії з використанням губчастої структури	181

УДК 005.21:005.8:004.8

Мандрик А.І., Лисенко С.М.

*Хмельницький національний університет***МЕТОД ОПТИМІЗАЦІЇ ПЛАНУВАННЯ ПРОЕКТІВ ТА ФОРМУВАННЯ КОМАНД З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ**

Розроблено метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму. Метод включає в себе розробку генетичного алгоритму та послідовність етапів, таких як ініціалізація популяції, мутація, кросовер, задля вибору оптимального розподілу розробників на проекті.

The methodological and practical aspects of creating a method for optimizing project planning and team formation using a genetic algorithm has been developed. The method involves the development of a genetic algorithm and a sequence of steps, such as population initialization, mutation, and crossover, to select the optimal distribution of developers on a project.

Вступ. В епоху швидкого розвитку та вдосконалення технологій, проблема невдалих програмних проектів стає все більш актуальною та вимагає термінового аналізу. Згідно з звітом CHAOS, опублікованим Standish Group у 2020 році, лише 16% програмних проектів виконуються успішно, дотримуючись запланованих термінів, бюджетів та стандартів якості [1]. Враховуючи ці обставини, невідкладним стає пошук шляхів оптимізації методів управління проектами, зокрема, в аспектах планування та формування команд. Важливість правильного технічного оснащення команди, а також раціональність планування та бюджетування проекту, не можна недооцінювати, адже вони можуть визначати, чи стане проект успішним, чи призведе до збитків [2-4].

Метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму. Метою є розробка методу, яка використовує генетичні алгоритми для оптимізації планування проектів та кадрового планування. Він має на меті забезпечити точне моделювання математичної природи планування проектів та формування команд за її допомогою [5-7]. Особлива увага приділяється визначенню відносних переваг різних можливих рішень відповідно до важливих якостей, що стосуються проблем планування та формування команд.

В рамках цієї роботи, методи генетичних алгоритмів дозволяють визначити оптимальні стратегії для відбору, кросинговеру та мутації, забезпечуючи максимізацію пристосованості протягом наступних поколінь [8, 9]. Основні області застосування цього методу включають кадрове забезпечення команд та планування розкладу проекту. Так, інформація про обмеження щодо навичок і послідовність

виконання завдань моделюються для забезпечення ефективного призначення завдань працівникам на основі їхніх навичок та досвіду.

Генетичний алгоритм, використовує гібридне кодування [10], яке дозволяє ефективно оптимізувати виконання завдань та призначення працівників. Час виконання завдань представлений додатковим, ненульовим цілим числом, що вказує, коли задача починається. Також, використовується двійкове кодування для визначення, чи був працівник задіяний у виконанні завдань (рисунк 1).

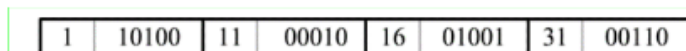


Рисунок 1 – Двійкове представлення генетичного алгоритму

Алгоритм був запущений 30 разів для кожного тестового проекту. В результаті, були отримані різні ступені збіжності. У першому експерименті близько 60% реалізацій виявились найкращими рішеннями, а в третьому експерименті – лише 5% збіжність була досягнута. Це підкреслює важливість стохастичності початкового вибору популяції та генетичних операцій у впливі на збіжність алгоритму [2].

Таблиця 1. Рівень досвіду співробітників для тестових проектів

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
S1	0	0.2	0	0	0	0	0	0	0	0.6
S2	0	0	0.8	0	0.6	0.6	0	0.6	0.2	0.4
S3	0.4	0.4	0	0.4	0	0.4	0.4	0.6	0.4	0
S4	0.8	0	0	0.8	0	0.8	0.8	0.8	0	0
S5	0	0	0	0.6	0	0	0	0	0	0.6

Основними факторами, які впливають на результати, є варіабельність часу виконання та стохастичність вибору (Рисунок 2).

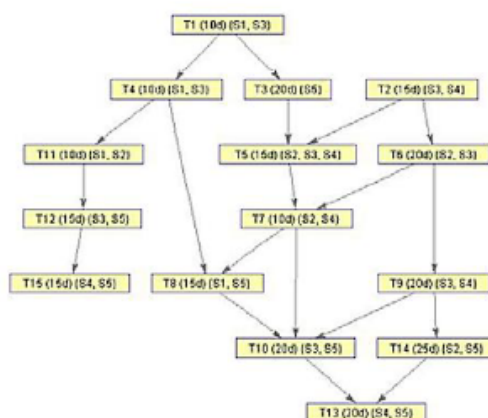


Рисунок 2 – Граф пріоритетів завдань для тестових проектів

Даний генетичний алгоритм використовує гібридне кодування, де час виконання завдань виражається через додатні цілі числа, а участь працівників – через двійкове кодування.

Експерименти

У цій роботі було використано метод на основі генетичного алгоритму, реалізований за допомогою бібліотеки DEAP та мови програмування Python [11].

Початковий експеримент полягав у виконанні шести різних завдань, кожне з яких вимагало певних навичок. Було вирішено об'єднати ці завдання з командою, що складалася з шести членів, кожен з яких мав унікальну комбінацію навичок. Щоб забезпечити всебічне дослідження простору рішень, генетичний алгоритм був налаштований на ітерації протягом 40 поколінь, кожне з яких містило популяцію з 50 потенційних командних утворень. З ймовірностями кросинговеру та мутації, зафіксованими на рівні 0,7 та 0,2 відповідно, алгоритм мав широкі можливості як комбінувати успішні рішення, так і вводити випадкові варіації.

У міру розгортання ітерацій алгоритм демонстрував чітку траєкторію вдосконалення. До кульмінації 40-го покоління він визначив оптимальний склад команди, який отримав оцінку придатності 6.0. Цей показник свідчив про ідеальну узгодженість, коли навички кожного члена команди резонували з вимогами поставленого перед ним завдання. Наприклад, оптимальне рішення отримала учасниця команди, наділена дизайнерськими навичками, Аліса, яка отримала завдання "Дизайн UI/UX", що було влучно позначено як "Дизайн UI/UX", результат роботи програми у консолі показано на рисунку 1.

```
me $
gen      nevals
0       50
1       31
2       42
... [intermediate generations and evaluations]
39      36
40      29

Best individual: ['Alice', 'Bob', 'Charlie', 'Dana', 'Emily', 'Frank']
me $ Fitness: 6.0
```

Рисунок 3 – Результат роботи алгоритму

Висновки

Розроблений метод з використанням генетичного алгоритму оптимізації дозволяє ефективно розв'язувати проблеми планування програмних проектів та

формування команд. Проблеми виникли при визначенні пріоритетності призначення досвідчених працівників, що мали вплив на загальну тривалість проекту. Вдосконалення, такі як застосування багатоцільової оптимізації та ітераційні уточнення, можуть виявитися корисними у розв'язанні цих викликів.

Перелік посилань

1. Standish Group CHAOS Report. Standish Group International. URL: <https://hennyportman.files.wordpress.com/2021/01/project-success-qrc-standish-group-chaos-report-2020.pdf> (дата звернення: 20/09/2023).
2. José G. M. Esgario, Iago E. da Silva, and Renato A. Krohling. Application of Genetic Algorithms to the Multiple Team Formation Problem. Arxiv, 2019.
3. Leksakul K., and Phetsawat S. Nurse scheduling using genetic algorithm Math. Probl. Eng. 2019 0–17 pp.
4. Alharbi A., Alqahtani K. A. Genetic Algorithm Solution for the Doctor Scheduling Problem The Tenth International Conference on Advanced Engineering Computing and Applications in Sciences. 2022. 91 p.
5. Kristiadi D., and Hartanto R. Genetic Algorithm for lecturing schedule optimization. 2019. 13–83 pp.
6. Parera S., Sukmana H. T., Wardhani L. K. Application of genetic algorithm for class scheduling. 2019. 3–7pp.
7. Mardiyono A. Intelligent System for Course Scheduling in Higher Educations Int. J. Inf. Technol. Bus. Manag. 2022. 29–34 pp.
8. Eletrotécnica E. Dynamic Scheduling for Maintenance Tasks Allocation supported by Genetic Algorithms. 2018. pp.
9. Kaleeswaran A., Ramasamy V., Vivekanandan P. Dynamic Scheduling Of Data Using Genetic Algorithm In Cloud Computing Int. J. Adv. Eng. 2018.
10. Swarm-based hybrid optimization algorithms: an exhaustive analysis and its applications to electricity load and price forecasting. URL: <https://link.springer.com/article/10.1007/s00500-023-07928-0> (дата звернення: 20.10.2023)
11. DEAP is a novel evolutionary computation framework for rapid prototyping and testing of ideas. URL: <https://deap.readthedocs.io/en/master/> (дата звернення: 10.08.2023)

ДОДАТОК В


Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів

НАУКОВИЙ КЕРІВНИК: Д. Т. Н., ПРОФЕСОР ЛИСЕНКО С. М.

ДОПОВІДАЧ: МАНДРИК А. І.

Хмельницький 2023

Мета, Об'єкт, Предмет дослідження

- Об'єктом дослідження є процес оптимізації планувальна роботи команд розробників проєктів.
 - Предметом дослідження є метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.
 - Метою кваліфікаційної роботи є підвищення ефективності роботи планувальника роботи команд розробників ПЗ із застосуванням апарат еволюційних алгоритмів.
- 

Мета і задачі дослідження

- Для досягнення поставленої мети необхідно вирішити такі завдання:
- дослідити методи та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів
- дослідити проблему оптимізації планування;
- проаналізувати відомі засоби та методи оптимізації;
- обґрунтувати вибір теоретичних та експериментальних методів для дослідження;
- підготувати дані для обробки генетичним алгоритмом;
- створити структуру інформаційної технології;
- розробити удосконалений метод інтелектуалізованого планувальника роботи команди розробників проєктів;
- провести експериментальне дослідження методу;
- розробити архітектуру інтелектуалізованого планувальника роботи команди розробників проєктів;
- скласти вимоги до інтелектуалізованого планувальника роботи команди розробників проєктів;
- провести оцінку ефективності розробленого алгоритму.

Наукова новизна

- Набув подальшого розвитку метод побудови інтелектуалізованого планувальника роботи команди розробників проєктів, який на відміну від відомих базується на оптимізаційних процесах, які ґрунтуються на застосуванні еволюційних алгоритмів, і уможливорює підвищити ефективність роботи планувальника роботи команд розробників ПЗ.
- Набули подальшого розвитку засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів.

Практичне значення

Практичне значення роботи полягає в тому, що запропонована інформаційна технологія інтелектуалізованого планувальника роботи команди розробників проектів може бути використаний як складова частина проектів де необхідний розподіл робочих процесів між працівниками.


Отримані результати можуть бути використані у розробці програмного забезпечення, для оптимізації процесів в команді при реалізації проектів різного призначення.



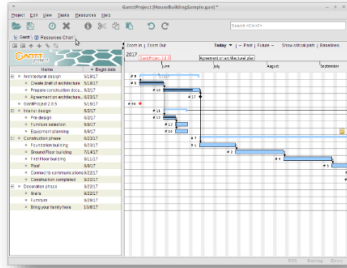
Актуальність

У сучасному динамічному світі, де технології та бізнес-вимоги швидко змінюються, ефективне управління проектами є ключовим для успіху будь-якої організації. Центральним аспектом управління проектами є оптимізація планування та розподілу ресурсів, що безпосередньо впливає на продуктивність команди та якість кінцевого продукту. Ця робота набуває актуальності, оскільки вона зосереджується на розробці та вдосконаленні інтелектуального планувальника робіт, що є важливим інструментом в сучасному управлінні проектами.

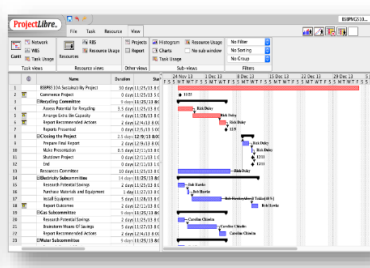
Інтеграція інформаційних технологій та використання еволюційних алгоритмів для оптимізації планування відкриває нові горизонти в ефективності та гнучкості управління проектами. Дослідження дозволяє краще зрозуміти, як сучасні технології можуть сприяти підвищенню продуктивності команд та якості розробки проектів, а також дозволяє розробити новітні методики та інструменти для цієї мети.



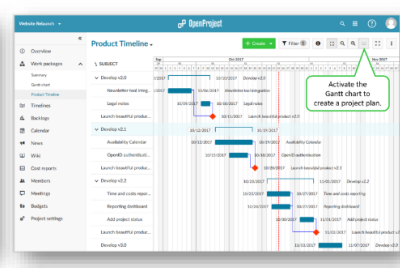
Відомі ринкові рішення



GanttProject



ProjectLibre



OpenProject

Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів

Для досягнення мети було розроблено удосконалений метод інтелектуалізованого планувальника роботи команди розробників проєктів.

При розробці вдосконаленого методу для інтелектуального планувальника роботи проєктної команди підхід зосереджується на структурованій і методичній обробці даних, що гарантує, що управління проєктом буде ефективним і адаптованим до різних вимог.

Вдосконалений метод інтелектуального планувальника роботи проєктної команди зосереджений на структурованій та методичній обробці даних, що забезпечує ефективне та адаптивне управління проєктами. Цей підхід є прикладом технологічного прогресу, спрямованого на оптимізацію планування та управління командами розробників проєктів.

Етап 1 : збір та попередня обробка вхідних даних

З когнітивної точки зору, навички, необхідні для виконання кожного завдання, повинні бути визначені і конкретизовані. Це вимагає ретельного вивчення кожного завдання для визначення точних наборів навичок, необхідних для його успішного виконання. Замість того, щоб просто перерахувати навички, заглиблюємося в кожну з них, щоб визначити необхідний рівень експертизи та спеціалізації.

Переходячи до попередньої обробки даних, на цьому етапі необхідно перетворити зібрані необроблені дані в структурований і чистий формат, придатний для аналізу та оптимізації з боку ГА.

Початковий крок проєкту передбачає нормалізацію наборів навичок. Це передбачає категоризацію та стандартизацію різноманітних навичок, перелічених для кожного завдання та члена команди.

До трансформації	Після трансформації
"Навички проєктування"	Навичка: 3
10 днів	Тривалість: 10
Високий пріоритет	Пріоритет: 1
Член команди А	Учасник: 1



Етап 2 : застосування генетичного для завдання оптимізації

Первинне представлення рішення - це хромосома, послідовність, де кожен ген представляє завдання і призначеного члена команди. Можна уявити, що це список, в якому кожна позиція - це завдання і член команди, відповідальний за нього.

Для оцінки придатності перш ніж вирішити, які рішення повинні "вижити" і "розмножитися", потрібно дізнатись, наскільки вони хороші. Саме тут у гру вступає фітнес-функція. Вона оцінює кожну хромосому на основі того, наскільки добре члени команди

ID учасник	Ім'я учасника	Навички	Доступність (дні/місяць)
M1	Тетяна	Дизайн, HTML, CSS	20
M2	Анна	Python, бази даних	18
M3	Микола	JavaScript, HTML, CSS	22
M4	Андрій	SQL, бази даних	19
M5	Сергій	Swift, дизайн	20
M6	Іван	Тестування, Java, Android	21

Етап 2 : застосування генетичного для завдання оптимізації

Цей метод використовує однокривий кросингвер. Для двох батьків C_1 і C_2 обирається випадкова точка кросингверу. Гени до цієї точки беруться від C_1 , а гени після - від C_2 для формування першого нащадка, і навпаки для другого нащадка. Для батьків C_1 та C_2 , якщо точка перетину k , тоді:

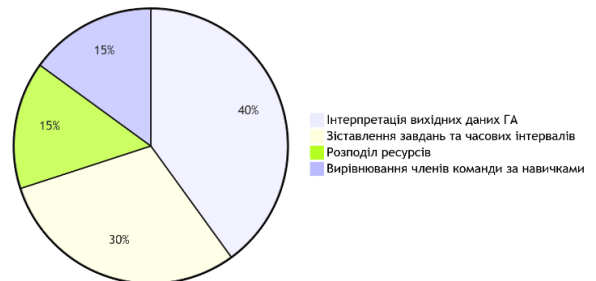
У результаті отримуємо хромосому з найвищим показником фітнесу. Та оцінку найкращого рішення.

ID завдання	ID призначеного учасника	Ім'я	Фітнес-внесок
T1	M5	Тетяна	0.92
T2	M2	Анна	0.85
T3	M3	Микола	0.89
T4	M4	Андрій	0.87
T5	M1	Сергій	0.9
T6	M6	Іван	0.88

Етап 3 : відображення та інтерпретації даних

Заключний етап розробки інтелектуального планувальника для проєктних команд включає в себе як відображення даних, так і виведення результатів. На цьому важливому етапі оптимальні рішення, створені генетичним алгоритмом, перетворюються на розклади, які можна втілити в життя. Цей етап є результатом після попередніх кроків, від збору даних до попередньої обробки та оптимізації ГА, і є ключовим для реалізації теоретичної роботи на практиці.

ГА видає результати, які кодують оптимальну послідовність і розподіл завдань і ресурсів. Щоб застосувати ці результати, необхідно їх розшифрувати. Наприклад, якщо ГА вказує послідовність завдань, наприклад, T3-T1-T2-T4, це означає запропонований порядок виконання завдань. Аналогічно, закодований розподіл ресурсів має бути перекладений на конкретні завдання для членів команди



Експериментальні дослідження методу

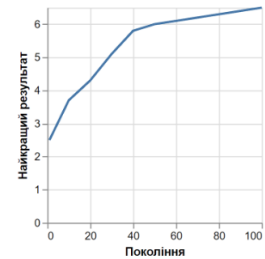
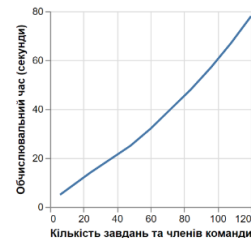
З метою оцінки роботи розробленого методу було проведено ряд експериментів.

Результат роботи алгоритму

```

gen 0 nevals
0 50
1 31
2 42
... [Intermediate generations and evaluations]
39 36
40 29
Best individual: ['Alice', 'Bob', 'Charlie', 'Dana', 'Emily', 'Frank']
Fitness: 6.0
  
```

Помітною перевагою алгоритму є його ефективність. Навіть коли складність проблеми зростала, при десятикратному збільшенні кількості завдань і членів команди, алгоритм незмінно знаходив якісні рішення за відносно короткий проміжок часу. У порівнянні з традиційними, ручними методами формування команд, перевага нашого алгоритмічного підходу була очевидною.



Експериментальні дослідження методу

Розроблений метод демонструє значне покращення часу до збіжності у порівнянні з традиційними інструментами управління проектами. Зокрема, він випереджає GanttProject на 3,33%, ProjectLibre на 2,27% та OpenProject на 2%. Таке прискорення у досягненні оптимального складу команди та розподілу завдань має вирішальне значення в динамічних проектних середовищах, де швидке прийняття рішень може суттєво вплинути на загальний успіх проекту.

Порівняно з GanttProject, він зменшує кількість помилок на 2,67%, порівняно з ProjectLibre - на 5,33%, а порівняно з OpenProject - на 3%.

Також розроблений метод має відмінну масштабованість, оскільки може ефективно управляти зростаючою кількістю завдань. Він перевершує GanttProject на 6,67%, ProjectLibre на 4,86% та OpenProject на 2,5%. Ця чудова масштабованість гарантує, що метод залишається ефективним і результативним зі зростанням складності проекту, уникаючи поширених пасток лінійного збільшення потреб у ресурсах, з якими часто стикаються традиційні інструменти.

Висновки

У кваліфікаційній роботі було проведено ряд теоретичних та практичних досліджень, присвячених удосконаленню метода, а також реалізації інтелектуалізованого планування роботи команд розробників проектів.

Висновки

У першому розділі було розглянуто існуючі проблеми та виклики у сфері планування роботи команд розробників, а також представлено основні підходи до розв'язання цих проблем.

У другому розділі було зосереджено увагу на вивченні та аналізі генетичних алгоритмів, їх застосування для оптимізації планування проектів та формування команд. Розроблено підходи до обробки та аналізу даних, що дозволяють підвищити ефективність планування.

Третій розділ охоплює розробку та впровадження удосконаленого методу планування роботи, заснованого на генетичних алгоритмах. Було продемонстровано, як цей метод може бути застосований для вирішення конкретних завдань планування та управління проектами, забезпечуючи оптимальний розподіл завдань та ресурсів серед членів команди. У четвертому розділі була представлена детальна архітектура та функціональні вимоги до інтелектуалізованого планувальника роботи.

Публікації з матеріалами дипломної роботи

- Мандрик А., Лисенко С. Метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму. Актуальні проблеми комп'ютерних наук АПКН-2023: зб. праць XV Всеукраїнської науково-практичної конференції, Хмельницький, м.Хмельницький, ХНУ, 17-18 листопада 2023. С. 177–183.
- Мандрик А., Лисенко С. Метод оптимізації планування проектів та формування команд з використанням генетичного алгоритму. Вісник Хмельницького Національного Університету. 2023.

Дякую за увагу !

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016013195

Дата перевірки:
17.12.2023 09:36:22 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
17.12.2023 09:37:08 EET

ID користувача:
100005591

Назва документа: Мандрик_Метод та засоби інформаційної технології інтелектуалізованого планувальника ...

Кількість сторінок: 97 Кількість слів: 15407 Кількість символів: 123205 Розмір файлу: 1.74 MB ID файлу: 1015699688

2.05% Схожість

Найбільша схожість: 0.76% з джерелом з Бібліотеки (ID файлу: 1015694417)

1.68% Джерела з Інтернету 139 Сторінка 99

1.45% Джерела з Бібліотеки 49 Сторінка 100

0.33% Цитат

Цитати 4 Сторінка 101

Посилання 1 Сторінка 101

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 16

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилоч в документах: 9%

ID: 123541 Назва: ДП Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів Додано в БД: 2023-12-17 Автора: Мандрик А.І. Керівники: Лисенко С.М. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	106893	855	1511 (1%)	24 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Мандрик Андрій Ігорович

Тема: Метод та засоби інформаційної технології інтелектуалізованого
планувальника роботи команди розробників проєктів

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість листів креслень - ____; кількість сторінок записки 82

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано структуру інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів на основі генетичного алгоритму.

2. Висновок про відповідність роботи дипломному завданню _____
Кваліфікаційна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі розглянуто принципи та методики інтелектуалізованого планування в контексті розробки програмного забезпечення. Аналізуються відомі інструменти та підходи, що застосовуються в цій області. У другому розділі розроблено модель для оптимізації процесу планування робіт команди розробників. У третьому розділі запропоновано методику використання генетичних алгоритмів для ефективного розподілу завдань і ресурсів в проєктах. У четвертому розділі представлено архітектуру інтелектуалізованого планувальника, яка інтегрує різні аспекти планування та оптимізації в єдину систему.

4. Позитивні сторони роботи: Розроблена модель, методологія та інтелектуалізована система управління проєктами дозволяють оптимізувати процеси планування в команді розробників. Це забезпечує вищий рівень ефективності у розподілі завдань та ресурсів, враховуючи різні аспекти та вимоги проєктних завдань

5. Негативні сторони роботи: У першому розділі не було досліджено деяких відомих методів для оптимізації які також мають ряд переваг.

6. Оцінка графічного оформлення та пояснювальної записки роботи: - _____

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

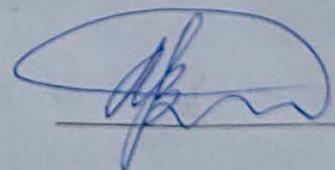
8. Інші зауваження: - _____

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «відмінно» 4,75 (А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Мартинюк Валерій Володимирович д.т.н., професор, завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки ХНУ.

“18” 12 2023р.



Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорущенко Т. О.

Мандрик Андрій Ігорович

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи ІСТМ-22-1

ЗАЯВА

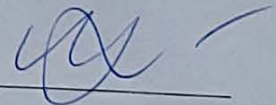
З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

19.12.2023

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів

Автор: _____ Мандрик Андрій Ігорович

Спеціальність: _____ 126 – Інформаційні системи та технології

Освітня програма: _____ Інформаційні системи та технології

Науковий керівник: _____ Лисенко Сергій Миколайович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах є збіг зі звітом з науково-дослідної практики автора Мандрика Андрія " Метод та засоби інформаційної технології інтелектуалізованого планувальника роботи команди розробників проєктів ", який було додано в репозитарій ХНУ 21 листопада 2023 року;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості Unicheck, складає 2.05% і адресується до 188 першоджерела; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____

Гарант ОП _____

Завідувач кафедри КІС _____

_____ С. М. Лисенко

_____ О. О. Павлова

_____ Т. О. Говорущенко